

الذكاء الإصطناعى

Artifitial-Intelligence

من الأساسيات وحتى النهايات

نسخة عربى

الفهرس

الفصل الأول: مدخل إلى الذكاء الصنعي

1. الذكاء

يتطلبُ فَهمُ الذكاء، باعتقادنا، فَهم كيفية اكتساب المعارف، وتمثيلها، وتخزينها، وكيف يجري توليد السلوك الذكي وتعلمه؛ كيف تتطور الدوافع والعواطف والأفضليات وكيف تُستخدم، وكيف تُحول الإشارات الحسية إلى رموز؛ وكيف تُعالَج الرموز لإجراء العمليات المنطقية، ومحاكمة ما جرى في الماضي، والتخطيط للمستقبل؛ وكيف تولد آلياتُ الذكاء ظاهرةَ الوهم، والاعتقاد، والأمل، والخوف، والأحلام – حتى اللطف والحب. وباعتقادنا، أن تفهم الأساس الذي تقوم عليه هذه الوظائف، سيكون إنجازاً علمياً من مستوى الفيزياء النووية ونظرية النسبية وعلم الوراثة الجزيئية.

- **الذكاء:** هو القدرة على الفهم وتعلم الأشياء.
- **الذكاء (في القاموس):** هو القدرة على التفكير والفهم عوضاً عن القيام بالأعمال بالفطرة أو آلياً.
- **الذكاء الطبيعي:** كائن له دماغ، أو شيء ما، يمكنّه من التعلم، والفهم، وحل المشاكل واتخاذ القرارات.

2. الذكاء الصنعي

علم يبحث في السلوك الذكي لغير الكائنات الحية.

يستلزم السلوك الذكي بدوره الإدراك **perception**، والمحاكمة **reasoning**، و التعلم **learning**، والتواصل **communicating**، والتصرف **acting** في بيئات معقدة.

أحد أهداف الذكاء الصنعي الطويلة الأمد هو إنشاء آلات يمكنها فعل هذه الأشياء بالجودة التي يقدمها البشر، أو حتى على نحوٍ أفضل من البشر. الغرض الآخر من الذكاء الصنعي هو فهم هذا النوع من السلوك سواء حدث من الآلات أم من البشر أم من حيوانات أخرى. ومن ثَم، يكون للذكاء الصنعي أهداف علمية وهندسية في آنٍ واحد.

يعتقد البعض أن الآلات المفكرة قد يجب أن تكون معقدة جداً، وأن تمتلك خبرات معقدة (مثل التفاعل مع بيئاتها ومع آلات مفكرة أخرى) بحيث لا يمكننا فعلياً تصميمها أو بناؤها. وتعطي الإجرائيات التي تولد الطقس العام مثالاً جيداً لذلك. فحتى لو علمنا كل ما هو هام عن الطقس فلن نستطيع بالضرورة أن نصنع نسخة مطابقة ِلهذه الظاهرة بكل غناها. لا يوجد نظام يستطيع صنع نسخة مطابقة ِلظاهرة الطقس بجميع تفاصيلها ويكون تعقيده أقلّ من سطح الأرض الفعلي ، وغلافها الجوي وبحارها ، الموجودة في الفضاء، وتدفئة الشمس لها، وتأثير المد والجزر فيها.

بالمشابهة، سيكون بناء ذكاء من مستوى ذكاء البشر على جميع الأصعدة، أمراً معقداً جداً، أو على الأقل معتمداً بشدة على الفيزيولوجيا الدقيقة للبشر، بحيث يتعذّر أن يوجد بصورة مستقلة عن تَجسده في بشر موجودين في بيئتهم.

لم يُبت بعدُ جواب السؤال الآتي:

هل سنتمكن فعلاً من بناء آلات مفكرة على مستوى البشر؟

وقد كان تقدم الذكاء الصنعي نحو هذا الهدف ثابتاً، وإن كان أبطأ مما توقع الرواد الأوائل. ونحن متفائلون فيما يتعلق بالنجاح النهائي.

أما تعريف الذكاء الصنعي، فهو يمكن أن يأخذ أحد الأشكال التالية، فيما يلي تعريفات اعتمدت له:

الذكاء الصنعي:

"جهود مثيرة لجعل الحواسيب (الآلات) تفكر.. أن يكون لها عقل بمعنى الكلمة"

(Haugeland 1985)

دراسة الإمكانات الذهنية عبر استخدام نماذج حسابية

(Charniak & McDermott 1985)

جزء من علوم الحاسوب يهتم بتصميم نظم حاسوبية ذكية تتصف بالخواص التي نصف بها البشر الأذكياء: فهم اللغات، التعلم، المحاكمة، حل المسائل، وغيرها...

(Barr & Feigenbaum (Jackson chapter 2))

3. مجالات الذكاء الصنعي

للذكاء الصنعي مجالات كثيرة ومتعددة، يبين الشكل أهمها، وتتضمن المجالات التالية:

- فهم اللغات الطبيعية المكتوبة ثم المنطوقة.
- تعرف الكلام وفهمه وتوليده حاسوبياً.
- الرؤية الحاسوبية وتعرف المشاهد واتخاذ قرارات.
- نظم التعليم الذكية بمعونة الحاسوب.
- الشبكات العصبونية.
- الربوطية ونظم التحـسس (رؤية وحركية ومعالجة إشارة للقيام بأفعال واتخاذ قرارات).
- الألعاب.
- برمجيات الترجمة وتلخيص المعلومات.
- البرمجة الآلية (برامج ذكية تنشئ أخرى)....

4. ما هو الحاسوب الذكي؟

- هل هو حاسوب لديه معطيات **Data**؟ أي لديه أرقام ومحارف وأسماء خام..؟ بالطبع لا، فتجميع المعطيات وخزنها ليس فيه كبير ذكاء.
- هل هو حاسوب لديه معلومات **Information**؟ أي إنه يتضمن نتائج معالجة رياضية وإحصائية لإعطاء توصيف مفيد للمعطيات، مثل إيجاد متوسط جميع القيم أو إيجاد القيمة الأعلى تكراراً أو القيمة الدنيا..؟ أيضاً لا، إذ لا تتطلب مثل هذه المعالجات أي ذكاء، وهي مجرد عمليات حسابية على المعطيات.
- هل هو نظام معلومات يشمل كيفية تخزين واسترجاع المعلومات؟ هنا أيضاً نتحدث عن قواعد معطيات فقط، وإن كان التوجه حالياً إلى نظم قواعد معطيات ذكية.
- هل هو نظام خبير؟ يجمع المعطيات، ينشئ معلومات، يستشير خبيراً عن صلاحية المعلومات والقواعد عليها، يضع برنامجاً لحل المسائل المتعلقة بهذه الخبرة؟ هنا فقط بدأنا نتحدث عن خبرة ومعرفة وذكاء.

بإمكاننا سرد عدة وجهات نظر حول أنماط الآلات التي يمكن أن تكون قادرة على التفكير كالبشر. ومن الادعاءات التي تُسمع غالباً:

- إن الدماغ يعالج المعلومات على التوازي، في حين تقوم الحواسيب التقليدية بذلك تسلسلياً. ومن ثَم علينا بناء مجموعة منوعة جديدة من الحواسيب المتوازية لتحقيق تقدم في الذكاء الصنعي.
- تعتمد آلية الحساب التقليدية على المنطق الاثناني **true** أو **false**. وعلى النظم الذكية فعلاً أن تستخدم نوعاً من "المنطق الترجيحي **fuzzy logic**".
- تعتبر العصبونات الحيوانية أكثر تعقيداً من المباديل **Switches** - وهي كتل البناء الأساسية للحواسيب. لذا سنكون بحاجة إلى استخدام عصبونات صنعية حقيقية في الآلات الذكية.

اقترح Turing اختباراً، سمي اختبار Turing، يمكّننا من الحكم على آلة معينة: أهي ذكية أم لا. في البداية، جرى وصف الاختبار كأنه لعبة. يشارك في اللعبة ثلاثة أشخاص، رجلٌ (A) وامرأة (B) ومستجوب Interrogator (C) يمكن أن يكون من أحد الجنسين. يبقى المستجوب في غرفة بعيداً عن الاثنين الآخرين (يتصل معهما بواسطة الكتابة من بُعد).

هدف اللعبة: أن يعين المستجوب أي من الشخصين الآخرين هو الرجل وأيهما المرأة. هو يعرفهما بلصيقتين **X** و**Y**، وفي نهاية اللعبة يقول إما "**X** هو **A** و**Y** هو **B**" أو "**X** هو **B** و**Y** هو **A**".

يُسمح للمستجوب أن يطرح على **A** و**B** أسئلة مثل:

C: أرجو من **X** إخباري طول شعره (أو شعرها) ؟

نفترض الآن أن **X** هو فعلاً **A**، يجب على **A** عندئذ أن يجيب. ودور **A** في هذه اللعبة أن يضلل **C** ليجعله يخطئ في التعرف.

ماذا يجري حين يكون أحد اللاعبين **A** و**B** آلة؟ غالباً ما يُبسط اختبار **Turing** إلى اختبار تحاول فيه الآلة إقناع مستجوب بشري أنها بشر.

جرى تقديم عدة إصدارات من هذا الاختبار (**Turing**) البسيط، ونظراً لأن بإمكان آلات بسيطة جداً أن تخدع المستجوب البشري مُدة، فلا يُعد الإصدار البسيط عادةً اختباراً فعالاً لذكاء آلة ما. على سبيل المثال، يستخدمُ برنامجُ **ELISA** الذي ابتكره **Joseph Weizenbaum** بعض الحيل البسيطة جداً (كما في المقتطفات الذي يظهرها الإيضاح)، ولكنه يبدو للمستخدم المتساهل قادراً على إجراء حوارٍ واقعي بعض الشيء، حتى لو كان لا معنى له. أما برنامج **JULIA** الذي ابتكره **Mauldin** فهو برنامج حوارٍ أكثر حداثةً وتعقيداً.

5. مقارنات هامة

ما الفرق بين الذكاء الطبيعي والذكاء الصنعي؟

الذكاء الصنعي دائم بمعنى أنه متى حصلنا على نظام خبير برمجي فهو يبقى لدينا، في حين لا نستفيد من خبرة وذكاء البشر إلا أثناء وجودهم بيننا.

الحصول على نسخ مكررة من النظام البرمجي الذكي ممكن وسهل المنال، في حين لا يمكن بسهولة نقل خبرة الخبير البشري إلى شخص آخر لنحصل على نسخة جديدة من خبرة الخبير.

يتطلب الحصول على الخبير البشري، في غالب الأحيان، تكلفة أعلى بكثير من الحصول على البرنامج الذكي.

الذكاء الصنعي متسق، بمعنى أننا في غالب الأحيان نستطيع أن نعلم على ماذا اعتمد البرنامج الذكي في اتخاذ قراراته، أما القرارات البشرية فلا نستطيع تفسيرها تفسيراً متسقاً ولا نستطيع التنبؤ بها . ولكن بالمقابل، الذكاء البشري خلاق ومبدع أما الصنعي فهو نمطي وموثق وليس فيه مفاجآت.

ما لفرق بين المعالجة الحاسوبية والمعالجة البشرية للمعطيات؟

مع أن المعالجة الحاسوبية هي أصلاً منتج بشري، إلا أن المعالجة البشرية غير الحاسوبية تختلف عن المعالجة الحاسوبية للمسائل، والجدول التالي يبين بعض هذه الفروق:

المعالجة البشرية	المعالجة الصنعية (الحاسوبية)

تستخدم رموزاً لتمثيل مفاهيم المسألة ومعالجتها بهدف اتخاذ قرارات	تستخدم أرقاماً، ومحارف
ثمة تمثيل مجرد للمعرفة[1]، وجداول[2] يجري تعلمها لحل المسائل	تستخدم الذاكرة والملفات لتخزين المعلومات والبرامج.
تستخدم برمجة منطقية وغيرها	تستخدم خوارزميات
تستخدم الكسبيات (التجريبيات **Heuristics**) والاستدلال ومواءمة الأشكال لاتخاذ قرارات أعقد ناجمة عن الخبرة	تنتج قرارات بسيطة...

ما الفرق بين الذكاء الصنعي والبرمجة التقليدية؟

المقصود هنا هو الفرق بين تقنيات الذكاء الصنعي وبرمجتها (استخدام لغات الذكاء الصنعي مثل **prolog**، **lisp**، **C++** الغرضية التوجه، **clips**، **Jess** وغيرها) والبرمجة التقليدية بلغات إجرائية (**C**، **Pascal**، **C#**،...). إننا لا ننسى أن تنفيذ لغات الذكاء الصنعي يستخدم لغات إجرائية، وبالنهاية يجري العمل مع الحواسيب التي تتعامل أصلاً مع بتات فقط نستخدمها لتمثيل مختلف المعطيات، إلا أن من وضع لغات الذكاء الصنعي وضع نصب عينيه فلسفة عمل مختلفة يبينها الجدول التالي:

الذكاء الصنعي	**البرمجة التقليدية**
يقوم بمعالجة رموز	تقوم بمعالجة حسابية
قد لا يكون الدخل والخرج معروفين تماماً في النظم الذكية	الدخل والخرج معرفان تماماً بالخوارزميات
البحث عن الحل تجريبي	ثمة خوارزميات للبحث
التركيز على المعرفة	التركيز على المعطيات والمعلومات
فصل التحكم عن المعطيات، إضافة معطيات جديدة مستقلة عن إضافة أدوات محاكمة وتحكم جديدة	المعطيات مدمجة مع أدوات التحكم، وأي تغيير يتطلب إعادة ترجمة البرامج قبل التنفيذ
سهلة التحديث نسبياً	صعبة التحديث للسبب السابق

6. هندسة المعرفة Knowledge Engineering

إن المعرفة هي المادة الأولية للذكاء الصنعي، وهي أعلى من المعطيات والمعلومات، إذ تتضمن الخبرة في مجال معين. على سبيل المثال، تقضي الخبرة في مجال كهرباء السيارات ، بفحص صلاحية بطارية السيارة أولاً حين لا يُدار المحرك.

[1] أصوات، صور، رموز،...
[2] مثال: جداول الضرب، طريقة القسمة،...

نطاق المعرفة

هي معرفة الخبير التي يستخدمها لحل مشاكل في هذا النطاق. غالباً يكون نطاق المعرفة جزءاً من نطاق المسألة، وقد لا يعرف الخبير كل شيء في نطاق المسألة. قد تكون الخبرة سطحية وقد تكون عميقة.

- **المعرفة السطحية:**

تتكون من قواعد تجريبية تصف الربط بين مختلف الأغراض في نطاق المعرفة، وهي عملياً ما يطبقه الخبير في معظم الأحيان. إنها معرفة تجريبية وغير مؤكدة دائماً.

- **المعرفة العميقة:**

وهي تتعلق بالمبادئ والقوانين الأساسية التي تتحكم في النطاق، ففي المثال الذي ذكرناه أعلاه عن الخبرة في كهرباء السيارات، المعرفة العميقة تتطلب معرفة السبب الذي يجعل البطارية مسؤولة عن العطل السابق.

ليست المعرفة العميقة أفضل من السطحية، ولكل منهما أهميته. يستخدم الخبير المعرفة السطحية للمسائل العامة، والمعرفة العميقة للمسائل الأعقد حيث لا تكفي المعرفة السطحية، أو حين لا تكون المعرفة العميقة أصلاً موجودة. لاسيما وأن الحصول على المعرفة العميقة أصعب، وكذلك تمثيلها.

ما هي هندسة المعرفة؟

- هي تحصيل المعرفة في مجال ما، من مصدر أو أكثر غير إلكتروني، وتحويلها إلى شكل يمكن استخدامه في الحاسوب، لحل مسائل يجري حلها عادة من قبل أشخاص ذوي معرفة كثيفة في هذا المجال. وقد بُذلت جهود كبيرة لتطوير كميات كبيرة من المعرفة الفعالة.
- يمكن ألا تكون المسألة التي نريد حلها معرفة تماماً. فقد لا نستطيع تصميم حل منذ البداية، وقد نضطر لتغيير تمثيل المعرفة مع تغير مهام المسألة.
- قد لا يكون للمسائل حل واضح لفحص النظام، ونتساءل هل توجد حلول معيارية؟
- يمكن أن يكون مجال المسألة معقداً جداً، ويتطلب تجميع كم كبير من المعرفة، و هذه المعرفة هي في الغالب غير دقيقة، وغير معرفة تماماً، وغير مفهومة جيداً. ويصعب في الغالب تحويلها إلى شكل حاسوبي.

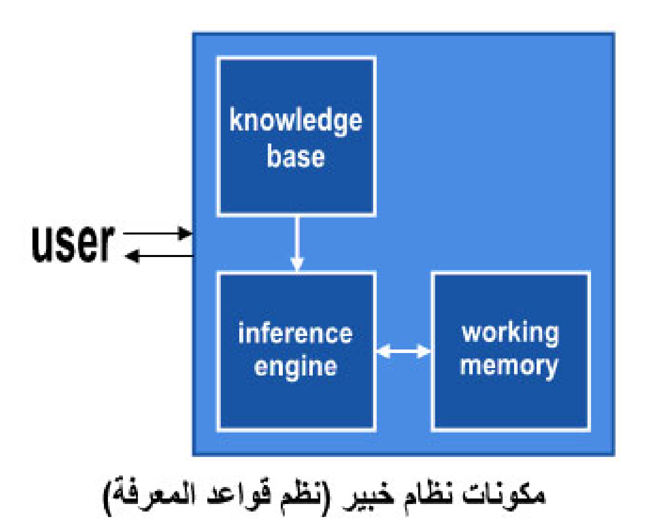
طرائق هندسة المعرفة

- **تحقيق نمذجة سريعة:** وذلك
  - بتطوير سريع لنموذج عمل **prototype**.
  - والتعلم من الأخطاء في بناء النماذج اللاحقة
  - والقيام بمراجعة دورية للنظام ولقاعدة المعرفة بناء على الاختبارات التي نجريها على النظام

➢ **اتباع منهجية مهيكلة:**

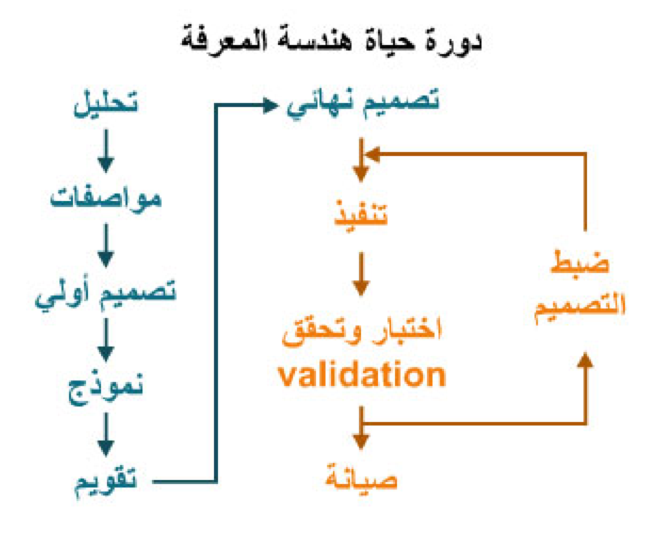
بحيث نعرف في كل لحظة، ما قمنا به، وأين نحن من هدفنا، وكيفية تحقيق الهدف. وبهذا تكون دورة حياة هندسة المعرفة كما في الإيضاح.

وبالنتيجة:



مكونات نظام خبير (نظم قواعد المعرفة)

- صيانة وتطور نظم قواعد المعرفة نشاط مفتوح لا ينتهي.
- النظم الخبيرة لا تعتمد على الخوارزميات وإنما على المعرفة. قد تتغير المعرفة القديمة ، و قد نحصل معارف جديدة، إذن أداء النظم الخبيرة في تحسن مستمر.
- تشير نظم قواعد المعرفة في غالب الأحيان إلى النظم الخبيرة. والنظام الخبير هو اجتماع قاعدة معرفة **knowledge base**، ومحرك استدلال **inference engine** يستخدم هذه المعارف للمحاكمة عليها ولاستنتاج معارف جديدة. يُضاف إلى هذين المكونين واجهة تخاطبية للتفاعل مع المستخدم خبيراً كان أم مستثمراً ، وذاكرة عمل **Working Memory** مشتركة لجميع المكونات.



7. فلسفة تمثيل المعرفة

حين نود تمثيل المعرفة، علينا تذكر الأدوار الخمسة التي يمكن أن يلعبها التمثيل وهي:

1. إيجاد بديل عن المعرفة نفسها، يقوم مقامها، ويساعد كياناً معيناً على المحاكمة وتحديد النتائج

اعتماداً على التفكير عوضاً عن القيام بأفعال.

2. إبداع مفاهيم عامة في المعرفة **Ontologies** (كيف يجب أن ننظر إلى العالم؟).
3. اعتماد نظرية تجزيئية للمحاكمة الذكية (كيفية المحاكمة، معرفة ما يقننه (يجعله قانونياً) الاستدلال وما ينصح به).
4. إيجاد وسط للحسابات الفعالة: كيف يقود التمثيل إلى تنظيم المعلومات لتسهيل الاستدلال المنصوح به.
5. إيجاد وسط ليعبر الناس عن معرفتهم: لغة ليقولوا أشياء عن العالم حولهم.

كل من هذه الأدوار يقود إلى خواص هامة وربما مختلفة لتمثيل المعرفة. ويحدد كل من هذه الأدوار إطار عمل قد يكون مختلفاً من دور إلى آخر. يجب تحديد ما هو مهم، لالتقاط وتمثيل الغنى الموجود في العالم الطبيعي حولنا بغية تمثيله والمحاكمة عليه والقيام بأفعال مناسبة.

الدور الأول: إيجاد البديل عن المعرفة بهدف المحاكمة عليها

إن المحاكمة على المعرفة شيء داخلي يُطبق على أشياء خارجية. والسؤال الأول الذي يطرح نفسه هو:

"ما دلالة هذه المعرفة؟"

وثمة سؤال هام آخر هو:

"ما دقة تمثيل المعرفة وما قرب هذا التمثيل من الحقيقة؟"

لاسيما وأننا يمكن أن نمثل أشياء مادية (مثل عجلة دراجة، مفتاح، منضدة...)، أو أشياء مجردة (مثل أفعال، إجرائيات، اعتقاد...)

بما أن البدائل ليست دقيقة تماماً، أي أن الغرض لن يتساوى مع تمثيله، فمن المؤكد أن المحاكمة باستخدام التمثيل قد لا تكون صحيحة تماماً. في تمثيل المعرفة، نحذف أشياء ونضطر إلى إضافة أشياء أخرى. نود أن يقلل التمثيل ما أمكن من أخطاء المحاكمة باستخدام التمثيل.

على سبيل المثال: حين نعرف المنضدة على أنها شيء له شكل معين، نغفل أبعادها فقد تشترك عدة طاولات في الشكل وتكون أبعادها مختلفة (إحداها مصغّرة عن أخرى..)، ونغفل المادة المصنوعة منها...

الدور الثاني: إبداع مفاهيم عامة في المعرفة

بما أننا لسنا دقيقين في التمثيل، فلابد من اتخاذ قرارات حول ما يجب علينا رؤيته من العالم حولنا (المنضدة: هل هي سطح مستو؟ هل يمكن أن تحمل أشياء أخرى؟ هل تشكل عقبة في الطريق؟ هل لها شكل محدد؟...)؛ وأن نركّز على ما نعتقد أنه مهم ولازم للمهام المطلوبة منها.

حين نود أن نحاكم حول إمكان تحرك ربوط في ممر يوجد فيه منضدة، فما يهمنا هو أبعاد المنـضدة وإن كان وجودها يعيق حركة الربوط. وحين نحاكم حول إمكان اندلاع حريق في مكان المنضدة، يهمنا معرفة المادة المصنوعة منها. وحين نحاكم حول إمكان وضع مكعبات على المنضدة تهمنا وضعيتها ومدى ملاسة سطحها...

يمكن كتابة هذه المفاهيم بعدة لغات برمجة (مثل **Lisp, Prolog, CLIPS**...)، لا يهم بأي لغة؛ المهم هو ما تعنيه هذه المفاهيم.

اعتماداً على ما سبق قد نختار المنطق لتمثيل المعرفة، أو الأطر **Frames** أو قواعد الإنتاج (قواعد شرطية من الشكل **if, then**)، يجب اختيار طريقة التمثيل منذ البداية. يمكن اختيار أكثر من طريقة تمثيل للمشكلة نفسها. على سبيل المثال، في مسألة تشخيص الأمراض اعتمد النظام الخبير **MYCIN** قواعد الإنتاج، في حين اعتمد النظام الخبير **INTERNIST** على الأطر فعرف نماذج أولية لأمراض الكلى أو أمراض الكبد.

الدور الثالث: نظرية تجزيئية للمحاكمة الذكية

يتضمن التمثيل جزءاً من الاعتقاد أو النظرة التي دفعت إليه. وهذه النظرة هي جزء من الظاهرة المعقدة المتعددة الوجوه للمحاكمة الذكية.

مكونات هذه النظرة هي:

- المفهوم الأساسي لتمثيل الاستدلال الذكي،
- معرفة الاستدلال الذي يقننه التمثيل،
- ومعرفة الاستدلال الذي ينصح به التمثيل.

وهذه المكونات هي الجواب على الأسئلة الثلاثة التالية:

- **ما هي المحاكمة الذكية؟**

- **رياضياً:** هي التي تعتمد المنطق والاستنتاج، ويكون الحل اعتماداً على الفرضيات ، ثم الإسناديات، ثم لغة البرمجة المنطقية **PROLOG**.
- **نفسياً:** هي التي تقلد سلوك الإنسان (مثل حل المسائل بالتخطيط والوصول إلى الهدف ، والنظم الخبيرة).
- **حيوياً:** تصرف يكون استجابة لتحريض، اعتماداً على ربط متوازٍ لعدة معالجات بسيطة (مثل الشبكات العصبونية الصنعية **ANN**، والوكلاء **agents**).
- **إحصائياً:** هي الخضوع إلى قوانين الاحتمالات (مثل المحاكمة على معارف غير مؤكدة).
- **اقتصادياً:** هي نظرية المنفعة.

وقد شهد التاريخ **عظماء كثيرين** كانوا وراء استخدام الأوجه المختلفة للمحاكمة الذكية.

Mathematical Logic	Psychology	Biology	Statistics	Economics
Aristotle				
Descartes				
Boole	James		Laplace	Bentham, Pareto
Frege		Bernoulli	Friedman	
Peano				
	Hebb	Lashley	Bayes	
Goedel	Bruner	Rosenblatt		
Post	Miller	Ashby	Tversky	Von Neumann
Church	Newell	Lettvin	Kahneman	Simon
Turing	Simon	McCulloch, Pitts		Raiffa
Davis		Heubel, Weisel		
Putnam				
Robinson				
LOGIC	SOAR	CONNECTIONIS M	Causal networks	Rational agents
PROLOG	Knowledge-based systems	A-life		
	Frames			

Views of Intelligent Reasoning and Their Intellectual Origins

الدور الثالث: نظرية تجزيئية للمحاكمة الذكية

➢ الاستدلال الذي يقننه التمثيل؟ (ماذا يمكن أن نستنتج؟)

إنها **الاستنتاجات السليمة Soundness** مثل قاعدة مودس بوننز : الموت نهاية كل إنسان، وسقراط إنسان، إذا نستنتج أن نهاية سقراط الموت، ومع ذلك فنحن نستخدم أيضاً **الاستقراء Abduction** إذا كانت الحمى التيفية تؤدي إلى ارتفاع الحرارة، فربما نستقرئ الحمى التيفية من ارتفاع الحرارة - رغم وجود أسباب أخرى عديدة لارتفاع الحرارة. بعد المنطق نشأت الأطر التي تعطي استنتاجات قد تكون خاطئة ولكنها تعطي دائماً طباقاً واضحاً **matching**.

➢ الاستدلال الذي ينصح به التمثيل؟ (ماذا يجب أن نستنتج؟)

إن ما يمكن أن نستنتجه كثير جداً، فما هو المهم بين هذا الكم الكثير من الاستنتاجات؟

المنطق لا يعطي الاستنتاج المنصوح به. على سبيل المثال، باستخدام قاعدة مودس بوننز على قاعدة معرفة تضم حقائق، نحصل على مجموعة حقائق جديدة، قد لا نكون بحاجة إليها في الوصول إلى أهدافنا أو محاكمتنا.

لذلك نفكر في طرق جديدة لتمثيل المعرفة مثل الأطر **frames**، والنماذج المعينة للتصرف **stereotypes, scripts**. (نصف ماذا نفعل في حفل عيد ميلاد؟ في المطعم؟ في الحافلة؟...)

إنها التقاط المعارف البديهية البسيطة التي تصعب برمجتها!

ربما كان على المستخدم أن يخبر النظام ماذا عليه أن يفعل، أو أن يقوده في فعل ما هو صحيح، أو أن يبني إستراتيجية استدلال خاصة به.

الدور الرابع: إيجاد وسط للحسابات الفعالة

في النهاية علينا تنفيذ تمثيل المعرفة والمحاكمة عليها باستخدام حواسيب وآلات و عمليات حسابية ، فكيف ننظم المعلومات لتسهيل الحصول على الاستدلال الأفضل على هذه الحواسيب؟

الهدف من استخدام الأطر كان هذا:

نُرفق بكل إطار عدة أنواع من المعلومات، بعضها حول كيفية استخدام الإطار، وبعضها حول ما نتوقع أن يحدث لا حقاً، وبعضها عما يجب فعله إذا لم تصدق هذه التوقعات... لذلك نصنف الأطر بأنها تسمح بالحصول على محاكمات تصنيفية **Taxonomic**. وتسير شبكات الدلالة باتجاه إيجاد روابط بسيطة بين العقد. أما نظم قواعد المعرفة فتنطلق من الهدف (سلسلة خلفية) أو من الحقائق (سلسلة أمامية) لإيجاد المعارف اللازمة المفيدة.

يستبعد علماء مضمون المعرفة **epistemology** عموماً الاهتمام بالحسابات.

الدور الخامس: إيجاد وسط ليعبر الناس عن معرفتهم

العالم حولنا غني بالمعارف، كيف نخبر الحواسيب بمعرفتنا عن العالم، أو كيف نخبر بها شخصاً آخر على أقل تقدير؟

ما مدى ملاءمة تمثيل المعرفة لوسط التعبير؟ ما مدى عموميته؟ وما مدى دقته؟... ما مدى ملاءمته للتواصل في هذا الوسط؟ وللتعامل مع لغته؟ ما هي الأشياء التي يسهل التعبير عنها في لغته؟ وما هي الأمور التي يصعب أو يستحيل التعبير عنها في هذه اللغة؟

بناء على ما تقدم نستنتج النقاط التالية:

- تمثيل المعرفة هو اللغة التي يجب أن نعبر بها ونتواصل بها. لذلك يجب أن نتكلمها دون كبير عناء.
- تمثيل المعرفة هو إطار عمل يساعد على إيجاد وجهات نظر صريحة وهامة في التمثيل والبحث في روح التمثيل.
- يجب توضيح الفرق في التصميم الناتج عن التساهل مع هذه الروح عوضاً عن إجبارها وخرقها.
- ما هي الأفكار التي نود تمثيلها؟ كيف يمكن تنفيذها وباستخدام أية تقنية؟ وعلى أي حاسوب؟
- على الرغم من أهمية أدوات وتقنيات التمثيل، إلا إن فهم ووصف غنى العالم حولنا يجب أن يكون هو شغلنا الشاغل.

8. مراجع البحث

- "الذكاء الصنعي: رؤية جديدة" من منشورات الجمعية العلمية السورية للمعلوماتية.
- **Giarratano & Riley "Expert Systems: Principles and Programming", 3d Edition, 1998**
- **"What is Knowledge Representation" R. Davis et al. (on the web)**

1. استخدام المنطق في المحاكمة

يمكن صياغة بعض المعلومات الصعبة التمثيل بصيغة قيود على قيم السمات. ويمكن في كثير من الأحيان، استخدام هذه القيود، التي تمثّل معارف هامة عن العالم حولنا، لاستنتاج قيم سمات أخرى لا يمكن تحسسها مباشرة. تختلف مسألة استنتاج المعلومات المتعلقة بالحالة الراهنة **present** (باستخدام حسابات تعتمد على القيود الموضوعة على السمات).عن مسألة حساب الحالات المستقبلية **future** (باستخدام طرائق البحث.)التي تنتج من جراء أفعال. نسمي العملية الأولى " **المحاكمة** **reasoning**" والثانية "**التخطيط** **planning**".

نركز الآن على مسألة المحاكمة.

هناك العديد من التطبيقات التي تتطلب محاكمةَ قيمِ السمات. فقد، ثَبتَ مثلاً أن بالإمكان تمثيل وظائف العديد من النظم الفيزيائية، ومنها النظم البيولوجية باستخدام مجموعة مناسبة من السمات . وترمز القيود الموضوعة على هذه السمات القوانين الفيزيائية، أو غيرها، المتعلقة بالكائن الحي أو بالجهاز . بعدئذ يمكن استخدام المحاكمة لأغراض أخرى كتشخيص الأعطال الوظيفية في هذه النظم ؛ إذ يمكن استنتاج السمات المرتبطة بـ "**الأسباب** **causes**" من السمات المرتبطة بـ " **الأعراض** **symptoms**". يعتبر هذا المنهج العام أساساً لطائفة هامة من تطبيقات الذكاء الصنعي، المعروفة بـ " **النظم الخبيرة** **expert systems**".

مثال:

ليكن لدينا ربوط قادر على رفْع كتلة: إذا كانت هذه الكتلة قابلة للحمل (أي غير ثقيلة جداً)، و كانت الشحنة الكهربائية المخزنة في بطارية الربوط كافية، عندها ستتحرك ذراع الربوط إذا ما حاول رفع الكتلة التي يمسكها. يمكن تمثيل مختلف هذه الشروط باستخدام السمات الاثنانية القيمة **(TRUE, FALSE)** التالية:

(BAT_OK) x_1
(LIFTABLE) x_2
(MOVES) x_3

لتسهيل النقاش سنطلق على هذه السمات أسماء (معطاة بين القوسين) يسهل تذكرها.

لنفترض أن الربوط يتحسس قيمة **BAT_OK** (بقراءة مقياس)، وقيمة **MOVES** (بطريق محسات موضوعة على مفاصله)، إلا أنه لا يتحسس قيمة **LIFTABLE**. علماً بأن قيمة **LIFTABLE** تعتبر هامة، وضرورية للربوط في بعض المهام الأخرى التي عليه إنجازها.

نعرف، من خلال هذا الوصف، أنه إذا كان لكل من **BAT_OK** و**LIFTABLE** القيمة **1** فستكون قيمة **MOVES** تساوي **1** أيضاً. لذلك، إذا كانت قيمة **MOVES** تساوي **0** عند محاولة الربوط تحريك الكتلة،

فإننا نعرف أن قيمة **BAT_OK** أو **LIFTABLE** (أو كليهما) تساوي **0**. لكن، إذا كانت قيمة **BAT_OK** وفقاً للمحس تساوي **1**، فلا بد أن تكون قيمة **LIFTABLE** مساوية **0**.

إذا كان بإمكاننا أن نُجري المحاكمة هكذا، فلْنجعل الربوط يقوم بذلك أيضاً . نحتاج إذن إلى لغة **language** يمكن بها التعبير عن القيود المفروضة على السمات وعن قيمها بآن واحد، إضافة إلى آلية استدلال **inference** تمكننا من تحقيق المحاكمة المطلوبة . يقدم حساب الفرضيات **propositional calculus**، وهو أحد فروع الجبر البولياني، الأدوات الضرورية لذلك.

يكتب القيد في المثال المطروح باستخدام لغة حساب الفرضيات كما يلي: **BAT_OK** $\wedge$ **LIFTABLE** $\supset$ **MOVES**

حيث $\wedge$ تعني "و"، و$\supset$ تعني "يقتضي". ويمكن استخدام آليات مرافقة لِهذه اللغة، لاستخلاص نتائج من العبارات في اللغة.

- **يستلزم المنطق:**
  - لغة (لها تركيب نحوي لتحديد ما هو التعبير المسموح به في اللغة).
  - قواعد استدلال لمعالجة الجمل في اللغة.
  - دلالات لربط عناصر اللغة بعناصر بعض الموضوعات.

سندرس لغتين منطقيتين:

- أولاهما وأبسطهما وتسمى حساب الفرضيات **propositional calculus**؛
- وتسمى الثانية، وهي أكثر فائدة، حساب الإسناديات من المرتبة الأولى **first order predicate (FOPC) calculus**.

تسمح لغة حساب الفرضيات بعرض العديد من الأفكار الهامة المتعلقة بحساب الإسناديات من المرتبة الأولى بصورة أشد بساطة، لذلك سنعرض تلك اللغة أولاً.

2. حساب الفرضيات

اللغة

تتكون لغة حساب الفرضيات صورياً من المكونات التالية:

- **الذّرات atoms:** لدينا الذرتان T وF (أي صح **True** وخطأ **False**)، والمجموعة العدْودة وغير المنتهية من سلاسل محرفية تبدأ بحرف كبير **(**مثل: **P, Q, R,..., P1, P2, ON_A_B,...** وهكذا).
- **الروابط connectives:** وهي $\vee$، $\wedge$، $\supset$، $\neg$، والتي تسمى على التتالي "أو"، "و"، "يقتضي"، "لا". سنعطي لاحقاً لهذه الروابط معنى يرتبط بأسمائها؛ أما الآن، فهي مجرد رموز غير ذات معنى.

يسمى التركيب النحوي للصيغ الجيدة التركيب **well-formed formulas (wffs)** جملاً:

- أية ذرة هي صيغة جيدة التركيب، مثال: **P**، **R** و**P3**.
- إذا كانت ω_1 و ω_2 صيغتين جيدتي التركيب فإن كلاً من الصيغ التالية هي كذلك:

صيغ جيدة التركيب باستخدام ω_1 و ω_2:

$\omega_1 \vee \omega_2$ (تسمى فصلاً لِـ ω_1 و ω_2)

$\omega_1 \wedge \omega_2$ (تسمى عطفاً من ω_1 و ω_2)

$\omega_1 \supset \omega_2$ (تسمى اقتضاء)

$\neg\omega_1$ (تسمى نفياً لِـ ω_1)

من الأمثلة على صيغ أخرى جيدة التركيب:

$(P \wedge Q) \supset \neg P$

$P \supset \neg P$

$P \vee P \supset P$

$(P \supset Q) \supset (\neg Q \supset \neg P)$

$\neg\neg P$

- نسمي الذرات أو الذرات المسبوقة بإشارة ¬ حرفيات **Literal**. ونسمي ω_1 في $\omega_1 \supset \omega_2$ بمقدمة **antecedent** الاقتضاء و ω_2 نتيجةَ **consequent** الاقتضاء.
- ليس هناك صيغ أخرى جيدة التركيب، مثال: $\neg\neg\supset P$ ليست صيغة جيدة التركيب.

تعتبر الأقواس في بعض المراجع جزءاً من اللغة، ولكننا سنستخدمها هنا بطريقة بديهية وغير دقيقة . يمكن تعرفُ الصيغ الجيدة التركيب باستخدام تعريفها العودي. فمثلاً، لبرهان أن $(P \wedge Q) \supset \neg R$ هي صيغة جيدة التركيب، نلاحظ أولاً أن $(P \wedge Q)$ هي صيغة جيدة التركيب لأن **P** و**Q** كذلك. و$\neg R$ هي صيغة جيدة التركيب لأن **R** كذلك. ومن ثَم، فإن $(P \wedge Q) \supset \neg R$ هي صيغة جيدة التركيب.

قواعد الاستدلال

ثمة عدة طرق لتوليد صيغ جيدة التركيب بدءاً من صيغ أخرى، نسمي ذلك قواعد الاستدلال **rules of inference**. وهي هنا مجرد طريقة ميكانيكية لإنشاء صيغة جيدة التركيب انطلاقاً من صيغ أخرى.

وبذلك، يكون لقاعدة الاستدلال نموذجياً الصيغةُ التالية:

يمكن استنتاج الصيغة γ من الصيغة α (أو من الصيغتين α وβ).

وفيما يلي مجموعة من قواعد الاستدلال الشائعة:

- يمكن استنتاج الصيغة ω_2 من الصيغتين ω_1 و$\omega_1 \supset \omega_2$ (تُسمى هذه القاعدة مودس بوننس **modus ponens**).
- يمكن استنتاج الصيغة $\omega_1 \wedge \omega_2$ من الصيغتين ω_1 وω_2 (إضافة العطف ∧).
- يمكن استنتاج الصيغة $\omega_2 \wedge \omega_1$ من الصيغة $\omega_1 \wedge \omega_2$ (∧ تبديلي).

- يمكن استنتاج الصيغة ω_1 من الصيغة $\omega_1 \wedge \omega_2$ (حذف العطف $\wedge$).
- يمكن استنتاج الصيغة $\omega_1 \vee \omega_2$ إما من ω_1 أو من ω_2 (إضافة الفصل $\vee$).
- يمكن استنتاج الصيغة ω_1 من الصيغة $(\neg\omega_1)$ $\neg$ (حذف النفي $\neg$).

تُسمى سلسلة الصيغ الجيدة التركيب $\{\omega_1, \omega_2,..., \omega_n\}$ برهاناً **Proof** (أو استنتاجاً **deduction**) لِ ω_n من مجموعة من الصيغ الجيدة التركيب Δ، إذا وفقط إذا وجدت كل صيغة ω_i ضمن Δ، أو كان بالإمكان استنتاجها من صيغة أو (عدة صيغ) سابقة في السلسلة، باستخدام إحدى قواعد الاستدلال. إذا كان هناك برهان لِ ω_n من Δ، فإننا نقول إن ω_n هي مبرهنَة **theorem** للمجموعة Δ. ونستخدم هنا الترميز التالي للتعبير عن إمكان البرهنة على ω_n من Δ: $\Delta \models \omega_n$

مثال:

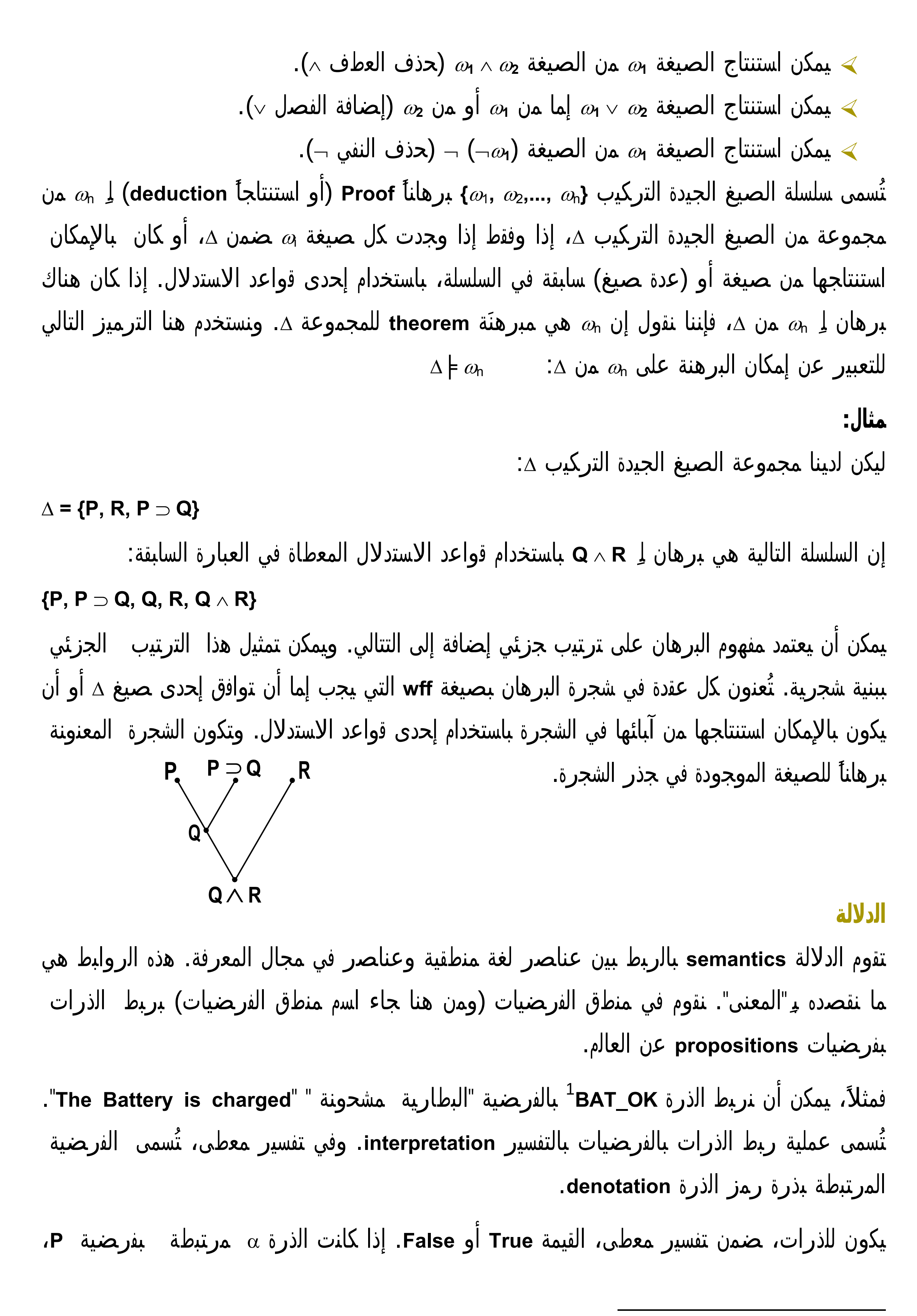
ليكن لدينا مجموعة الصيغ الجيدة التركيب Δ:

$\Delta = \{P, R, P \supset Q\}$

إن السلسلة التالية هي برهان لِ $Q \wedge R$ باستخدام قواعد الاستدلال المعطاة في العبارة السابقة:

$\{P, P \supset Q, Q, R, Q \wedge R\}$

يمكن أن يعتمد مفهوم البرهان على ترتيب جزئي إضافة إلى التتالي. ويمكن تمثيل هذا الترتيب الجزئي ببنية شجرية. تُعنون كل عقدة في شجرة البرهان بصيغة **wff** التي يجب إما أن توافق إحدى صيغ Δ أو أن يكون بالإمكان استنتاجها من آبائها في الشجرة باستخدام إحدى قواعد الاستدلال. وتكون الشجرة المعنونة برهاناً للصيغة الموجودة في جذر الشجرة.



الدلالة

تقوم الدلالة **semantics** بالربط بين عناصر لغة منطقية وعناصر في مجال المعرفة. هذه الروابط هي ما نقصده بِ"المعنى". نقوم في منطق الفرضيات (ومن هنا جاء اسم منطق الفرضيات) بربط الذرات بفرضيات **propositions** عن العالم.

فمثلاً، يمكن أن نربط الذرة **BAT_OK**[1] بالفرضية "البطارية مشحونة " "**The Battery is charged**".

تُسمى عملية ربط الذرات بالفرضيات بالتفسير **interpretation**. وفي تفسير معطى، تُسمى الفرضية المرتبطة بذرة رمز الذرة **denotation**.

يكون للذرات، ضمن تفسير معطى، القيمة **True** أو **False**. إذا كانت الذرة α مرتبطة بفرضية **P**،

[1] لسنا مضطرين إلى استخدام سلسلة محرفية سهلة التذكر، إذ يمكن الاستعاضة عنها بسلاسل محرفية أخرى.

نقول إن للذرة α القيمة **True** إذا كانت الفرضية **P** صحيحة في عالم المسألة، وإلاّ تكُنْ قيمتها **False**.

مثال: الذرة **Bat_Ok** تدل على فرضية كون البطارية بحالة جيدة، ولذا تأخذ هذه الذرة القيمة **True** حين تتحقق هذه الفرضية.إن قيمة الذرة الخاصة **T** هي دائماً **True**، وقيمة الذرة الخاصة **F** هي دائماً **False**. ولما كان من الضروري أن تكون فرضيات العالم إما صحيحة أو لا (وهذا غير واقعي دائماً!)، فبالإمكان تحديد قيمة تفسير بإسناد القيم مباشرة إلى الذرات في اللغة - بقطع النظر عن فرضية العالم التي ترمز إليها كل ذرة.

يعني الورود الصريح لعنصر ما (مثلاً **X1**) في قاعدة المعرفة أن الفرضية الموافقة صحيحة في عالمنا. سنرى لاحقاً كيفية استخدام الصيغ الجيدة التركيب في قاعدة المعرفة. يبين الجدول التالي دلالات الروابط في حساب الفرضيات:

ω_1	ω_2	$\omega_1 \wedge \omega_2$	$\omega_1 \vee \omega_2$	$\neg\omega_1$	$\omega_1 \supset \omega_2$
True	**True**	**True**	**True**	**False**	**True**
True	**False**	**False**	**True**	**False**	**False**
False	**True**	**False**	**True**	**True**	**True**
False	**False**	**False**	**False**	**True**	**True**

لنفترض أن **P** تأخذ القيمة **False**، و**Q** تأخذ القيمة **False** و**R** تأخذ القيمة **True**، وأننا نريد حساب قيمة الصيغة:

$[(P \supset Q) \supset R] \supset p$

بالحساب من الداخل باتجاه الخارج، نجد أولاً أن قيمة**$P \supset Q$** هي **True**؛ ثم نحسب قيمة **$(P \supset Q) \supset R$** فنجدها **True** أيضاً. ولما كانت **P** لها القيمة **False** كانت قيمة التعبير الكلي هي **False**.

نقول عن تفسير ما إنه يحقق صيغة جيدة التركيب إذا كانت قيمة هذه الصيغة وفق هذا التفسير هي **True**. ونقول عن تفسير يحقق صيغةً جيدة التركيب إنه نموذج **model** لها. ونقول عن تفسير يحقق كل صيغة جيدة التركيب في مجموعة من الصيغ الجيدة التركيب بأنه نموذج لهذه المجموعة.

يمكن ألا يكون هناك تفسير يحقق صيغة جيدة التركيب (أو مجموعة من الصيغ). نقول في هذه الحالة إن الصيغة (أو مجموعة الصيغ) غير متسقة أو غير قابلة للتحقيق. فالصيغتان **F** و**$P \wedge \neg P$** غير قابلتين للتحقيق (ليس هناك تفسير يجعل قيمة هاتين الصيغتين **True**). كذلك، فإن مجموعة الصيغ:

$\{P \vee Q, P \vee \neg Q, \neg P \vee Q, \neg P \vee \neg Q\}$

غير قابلة للتحقيق.

استخدم جدول الحقيقة ليتأكد لك أنه لا يوجد أي تفسير يجعل جميع صيغ هذه المجموعة صحيحة معاً.

نقول عن صيغة إنها صحيحة valid[2] إذا كانت قيمتها **True** في جميع تفاسير ذراتها المكونة. من الأمثلة على الصيغ الجيدة التركيب الصحيحة ما يلي:

$P \supset P$ (وهي تكافئ $\neg P \vee P$)
T
$\neg (P \wedge \neg P)$
$Q \vee T$
$[(P \supset Q) \supset P] \supset P$
$P \supset (Q \supset P)$

نقول عن صيغتين جيدتي التركيب إنهما متكافئتان**equivalent** إذا وفقط إذا كانت لهما قيم الحقيقة نفسها في جميع التفاسير. انظر التكافؤات التالية:

قانونا دمورغان **DeMorgan**:

$\neg (\omega_1 \vee \omega_2) \equiv \neg \omega_1 \wedge \neg \omega_2$
$\neg (\omega_1 \wedge \omega_2) \equiv \neg \omega_1 \vee \neg \omega_2$

قانون عكس الإيجاب **Contrapositive**:

$(\omega_1 \supset \omega_2) \equiv (\neg \omega_2 \supset \neg \omega_1)$

إذا كانت ω_1 وω_2 متكافئتين فإن الصيغة التالية صحيحة:

$(\omega_1 \supset \omega_2) \wedge (\omega_2 \supset \omega_1)$

وبسبب هذه الحقيقة، كثيراً ما تُستخدم الكتابة $\omega_1 \equiv \omega_2$ على أنه اختصارٌ لِ $(\omega_1 \supset \omega_2) \wedge (\omega_2 \supset \omega_1)$.

إذا كان لصيغة جيدة التركيب القيمة **True** في كل التفسيرات التي تجعل لكل صيغة جيدة التركيب في مجموعة Δ من الصيغ القيمة **True**، فإننا نقول إن Δ تستلزم منطقياً ω، وإن ω تتبع منطقياً Δ أو إن ω هي نتيجة منطقية لِ Δ. نستخدم الرمز $\models$ للدلالة على الاستلزام المنطقي، ونكتب $\Delta \models \omega$. مثلاً:

$\{P\} \models P$
$\{P, P \supset Q\} \models Q$
$F \models \omega$

حيث ω هي أية صيغة جيدة التركيب

$P \wedge Q \models P$

يستخدم الاستلزام المنطقي كثيراً في مسائل الذكاء الصنعي، إذ يقدم طريقة فعالة لإظهار أنه إذا كانـت مجموعة فرضيات عن العالم صحيحة، فإن فرضية هامة أخرى (ربما لا يمكن تحسسها) يجب أن تكون صحيحة أيضاً.

إذا تحقق أنه حين تبرهن المجموعة Δ بحسب قواعد الاستدلال على الصيغة ω، فإن هذا يقتضي أن Δ تستلزم ω؛ فإننا نقول إن مجموعة قواعد الاستدلال سليمة.

[2] إن الصيغة الجيدة التركيب الصحيحة تكون عديمة الأهمية؛ لأنها لا تُعطي أية معلومة عن حالة العالم.

وإذا تحقق أنه متى كان $\Delta \models \omega$ يوجد برهان لِـ ω من Δ باستخدام مجموعة قواعد الاستدلال، فإننا نقول إن مجموعة القواعد كاملة.

عندما تكون قواعد الاستدلال سليمة وكاملة، فيمكن أن نُحدد وجود أو عدم وجود تبعية صيغة ما جيدة التركيب تَبعيةً منطقية لِمجموعة من الصيغ جيدة التركيب، وذلك بالبحث عن برهـان لهـذه الصيغة عوضاً عن استخدام جدول الحقيقة.

عندما تكون قواعد الاستدلال سليمة، ونجد برهاناً لِـ ω من Δ، فإننا نعرف أن ω تتبع منطقياً Δ. وعندما تكون قواعد الاستدلال كاملة، فإننا نعرف أننا سنكون قادرين على تأكيد أن ω تتبع Δ (عنـدما تكون كذلك فعلاً) باستخدام إجراء بحث كامل للبحث عن برهان. إن الاستعاضة عن طرائق جدول الحقيقة بطرائق البرهان يجعل تكلفة الحسابات في حساب الفرضيات وحساب الإسناديات (الذي سندرسه لاحقاً) أقل بكثير.

الحل Resolution

لتكن Σ_1 وΣ_2 مجموعتين من الحرفيات وλ ذرة. يمكن من الصيغتين $\Sigma_1\cup\{\lambda\}$ و$\Sigma_2\cup\{\neg\lambda\}$ أن نستنتج أن $\Sigma_1\cup\Sigma_2$، وهذا ما يسمى ناتج حل العبارتين Σ_1 وΣ_2 (أو اختصاراً حلّ). وتُسمى الذرة λ المحلول، والإجرائية الحل.

فيما يلي بعض الأمثلة:

- يؤدي حل $\mathbf{P \vee R}$ و$\mathbf{\neg P \vee Q}$ إلى $\mathbf{Q \vee R}$. **كيف؟**

يمكن أن نكتب العبارات التي جرى حلها كاقتضاءين $\mathbf{\neg R \supset P}$ و$\mathbf{P \supset Q}$. عند تطبيق قاعدة الاستدلال المسماة السلْسلَة **chaining** على الاقتضاءين السابقين نحصل على $\mathbf{\neg R \supset Q}$، التي تُكافئ الحل $\mathbf{Q \vee R}$. ونرى، أن السلسلة هي حالة خاصة من الحل.

- يؤدي حل $\mathbf{R}$ و$\mathbf{\neg R \vee P}$ إلى $\mathbf{P}$: إذ كانت العبارة الثانية تُكافئ $\mathbf{R \supset P}$، نلاحظ أن مودس بوننس هي أيضاً حالة خاصة من الحل.
- يؤدي حل $\mathbf{P \vee Q \vee R \vee S}$ مع $\mathbf{\neg P \vee Q \vee W}$ على $\mathbf{P}$ إلى $\mathbf{Q \vee R \vee S \vee W}$: لاحظ أن $\mathbf{Q}$ تظهر مرة واحدة في عبارة الحل (والتي هي معرفة باعتبارها مجموعة أصلاً).
- يؤدي حل λ (وهو حرفي موجب) مع $\neg\lambda$ (وهو حرفي سالب) إلى العبارة الخالية. أي يمكننا من λ و$\neg\lambda$ استنتاج $\mathbf{F}$ لأن λ و$\neg\lambda$ متناقضان: إن أية مجموعة من الصيغ الجيدة التركيب تحوي λ و$\neg\lambda$ هي مجموعة غير قابلة للتحقيق.

يمكن تحويل أية صيغة في حساب الفرضيات إلى عطف من العبارات مكافئ لها. ونقول عن الصيغة المكتوبة كعطف من العبارات إنها من شكل العطف النظامي (كما نقول عن صيغة مكتوبة كفصل من

حرفيات معطوفة إنها من شكل الفصل النظامي).

مثال

سنستعرض في المثال التالي خطوات إجرائية تحويل صيغة ما جيدة التركيب إلى شكل العطف النظامي.

لإيضاح الإجرائية سنستخدم الصيغة التالية: $\neg (P \supset Q) \vee (R \supset P)$

1. حذف إشارات الاقتضاء وذلك باستخدام الشكل المكافئ مع $\vee$:

$\neg (\neg P \vee Q) \vee (\neg R \vee P)$

2. تضييق مجال إشارات النفي $\neg(P \supset Q)$ تصبح $(\neg P \vee Q\neg)$ باستخدام قانوني دومرغان، وبحذف إشارتي النفي المتتاليتين:

$(P \wedge \neg Q) \vee (\neg R \vee P)$

3. استخدام قوانين التجميع والتوزيع

$\neg(P \wedge Q) = (\neg P) \vee (\neg Q)$

$\neg(P \vee Q) = (\neg P) \wedge (\neg Q)$

للوصول إلى شكل العطف النظامي، فنحصل على:

$(P \vee \neg R \vee P) \wedge (\neg Q \vee \neg R \vee P)$

ثم على:

$(P \vee \neg R) \wedge (\neg Q \vee \neg R \vee P)$

نعبر عادةً عن عطف من العبارات بمجموعة من العبارات (العطف ضمني بين العبارات):

$\{ (P \vee \neg R) , (\neg Q \vee \neg R \vee P) \}$

الحل بالنقض

يتم الحل بالنقض لإثبات صحة صيغة **wff** من مجموعة من الصيغ Δ باتباع الخطوات التالية:

- تحويل كل صيغة، في Δ إلى عطف عبارات (شكل العطف النظامي).
- تحويل نفي الصيغة **w** المطلوب إثباتها إلى عطف عبارات.
- تجميع العبارات الناتجة عن الخطوتين السابقتين في مجموعة واحدة Γ.
- نعاود تطبيق الحل على عبارات المجموعة Γ ونضيف ناتج الحل في كل مرة إلى المجموعة ، حتى الوصول إلى حالة لا يبقى فيها عبارات قابلة للحل أو الوصول إلى المجموعة الخالية.

مثال

لتكن لدينا المعارف التالية:

(1) **BAT_OK** البطارية بحالة جيدة

(2) $\neg MOVES$ الربوط لا يتحرك

(3) $MOVES \supset LIFTABLE \wedge BAT_OK$ البطارية بحالة جيدة والربوط قادر على الحمل فإن الربوط سيتحرك.

يعطي تحويل العبارة الثالثة إلى الشكل النظامي:

(4) $\neg BAT_OK \vee \neg LIFTABLE \vee MOVES$

نريد أن نبرهن أن الغرض لا يمكن رفعه **LIFTABLE¬**، يعطي نفي الصيغة المطلوب برهانها العبارة التالية:

(5) **LIFTABLE**

لنطبق الآن الحل للحصول على العبارات التالية:

(6) بحل (4) و(5) نحصل على $\neg BAT_OK \vee MOVES$

(7) بحل (2) و(6) نحصل على $\neg BAT_OK$

(8) بحل (1) و(7) نحصل على **Nil**

نبين شجرة الحل في الشكل الجانبي.

LEFTABLE
¬BAT_OK ∨ ¬LIFTABLE ∨ MOVES
¬BAT_OK ∨ MOVES
MOVES
¬BAT_OK
BAT_OK
Nil

استراتيجيات الحل بالنقض

على الرغم من سهولة الحل بالنقض، ولكن يجب اختيار إستراتيجية تبين ماذا علينا أن نحل أولاً.

هناك استراتيجيات ترتيب الحل:

- الحل عمقاً أولا،
- أم عرضاً أولاً.

واستراتيجيات تشذيب Refinement:

- ضرورة احتواء الحل على نفي ما مطلوب برهانه أو عبارة خاف لها،
- أو ضرورة احتواء الحل على عبارة من المجموعة الابتدائية (قبل إجراء أي حل)،
- أو ضرورة تحقيق الشرطين السابقين.

عبارات هورن

هي العبارات التي تحتوي حرفياً موجباً واحداً على الأكثر ولها ثلاثة أنواع:

- العبارة الوحيدة التي تحوي ذرة وحيدة وندعوها حقيقة **Fact**، مثل **P** و**Q**.
- الاقتضاء (و الذي ندعوه عادةً قاعدة **Rule**). تتألف مقدمته من عطف حرفيات موجبة ونتيجته من حرفي موجب وحيد، مثل $P \wedge Q \supset R$.

- مجموعة من الحرفيات السالبة والمكتوبة على شكل اقتضاء مقدمته تتألف من عطف حرفيات موجبة ونتيجته المجموعة الخالية. يمكن الحصول على هذا الشكل مثلاً عند نفي صيغة **wff** مطلوب إثباتها تتألف من عطف حرفيات موجبة. ندعو هذه الصيغة عادةً بالهدف **Goal**، مثل **$\supset$P$\wedge$Q**.

وبذلك نرى أن حساب الفرضيات يسمح بتمثيل نظام خبير يعتمد قواعد الإنتاج، إذ يمكن تمثيل الحقائق والقواعد، والأهداف التي نود برهنتها. وتكون المحاكمة باستخدام الحل والبرهان بالحل بالنقض.

3. الإسناديات predicates

تعتبر لغة حساب الفرضيات محدودة. إذ لا يمكننا مثلاً التعبير عن حقيقة أنه عندما نحرك الكتلة **B** مثلاً، فإن **B** هي الكتلة نفسها التي تؤكد الفرضيةُ **ON_B_C** أنها موضوعة على الكتلة **C**. ففي حساب الفرضيات تكون الذراتُ سلاسل محرفية ليس لها بنية داخلية. والفرضيتان **ON_A_B** و**ON_B_C** في عالم الكتل مثلاً، مختلفتان تماماً وليس بينهما شيء مشترك. ومع أننا سمينا هذه الذرات بأسماء ذات دلالة (لتساعدنا على تذكر المقصود بها)، فلا شيء يمنعنا من استخدام أسماء أخرى لها مثل **P124** و**Q23**.

من المفيد استخدام لغة تدل على الأغراض في العالم (مثل الكتل)، وعلى الفرضيات المتعلقة بالعالم . أي إننا نحتاج إلى لغة فيها تسميات للأغراض التي نضع لها الفرضيات، وللفرضيات الموضوعة. ففي عالم الكتل مثلاً، يمكن أن تكون لدينا فرضيات مثل **ON_B_C $\supset$ $\neg$CLEAR_C**، حيث تعني **CLEAR_C** عدم وجود أي شيء على الكتلة **C** (أي إن وجود الكتلة **B** فوق **C** تقتضي ألا يكون سطح **C** خالياً). يلزمنا للتعبير عن حقيقة مثل هذه لجميع كتل العالم، الكثير من الفرضيات. لذا من المناسب أن يكون بإمكاننا مجرد قول شيء مثل **On (x,y)$\supset$$\neg$Clear (y)**، حيث **x** و**y** متغيران يمكن أن يعبرا عن أية كتلة.

اللغة

هي لغة تضم ثوابت غرضية إضافة إلى ثوابت علاقاتية وتابعية، لدينا ما يلي:

- مجموعة غير منتهية من الثوابت الغرضية **object constants**. سنستخدم لترميزها سلاسل محرفية، عادةً ما تكون لها دلالة تُساعدنا على تذكر مدلولها ، مثل : **Aa**, **13B**, **John**, **EiffelTower**.
- مجموعة غير منتهية من الثوابت الدالية **function constants** من جميع الأبعاد ، مثل :

distanceBetween(Damascus, Homs)
times (5, 4)

- مجموعة غير منتهية من الثوابت العلاقاتية **relation constants** من جميع الأبعاد (نسمي الثوابت العلاقاتية أحياناً الإسناديات)، من الأمثلة على الثوابت العلاقاتية : **Son(Radwan, Anas).**
- مجموعة من المتغيرات، مثل: **times (4, x)**
- مجموعة الروابط ∨، ∧، ¬، ⊃ و∀ و∃ والمحددات]، [،)، (والفاصلة ، .

يمكن استخدام حساب الإسناديات لتمثيل المعرفة، كما في المثال التالي:

- **Block (A)**
- **Block (B)**
- **Block (C)**
- **Floor (F1)**
- **On (B,A) , On (A,C)..**
- **Clear (B)**

B
A
C
Floor

حيث يبين الشكل التالي التقابل بين العالم وبين الإسناديات التي اتخذناها لتمثيله:

حساب الإسنادية	العالم
A	A
B	B
C	C
F1	Floor
On	On = {<B,A>,<A,C>,<C,Floor>}
Clear	Clear = {<B>}

قواعد الاستدلال

إن قواعد الاستدلال التي استخدمناها (مثل: مودس بوننس، وإدخال العطف وحذفه، وإدخال الفصل، وحذف النفي والحل) في حساب الفرضيات تبقى صحيحة ويضاف إليها قواعد أخرى نبينها فيما يلي:

- **الاستنساخ العام Universal Instantiation:** أي إنه من **(∀x)P(x,f(x),B)** نستطيع أن نستنتج **P(A,f(A),B)**، إذا تحققت إسنادية مهما تكن **x**، فهي تتحقق من مستنسخ **A** من **x**.
- **التعميم الوجـودي Existential Generalization:** أي إنه من **(∀x)Q(A,g(A),x)** نستنتج **(∃y)(∀x)Q(y,g(y),x).**

يمكن باستخدام دلالة المكميات المعطاة إثبات التكافؤات التالية، والمشابهة لقوانين دومورغان:

$\neg(\forall\xi)\ \omega(\xi) \equiv (\exists\xi)\ \neg\omega(\xi)$ إذا لم تكن صيغة محققة مهما يكن متحولها فهذا يعني أنه يوجد مثال لا تتحقق من أجله الصيغة.

$\neg(\exists\xi)\ \omega(\xi) \equiv (\forall\xi)\ \neg\omega(\xi)$ إذا لم يوجد مثال تتحقق الصيغة من أجله فهذا يعني أنها غير محققة مهما يكن متحولها.

$(\forall\xi)\ \omega(\xi) \equiv (\forall\eta)\ \omega(\eta)$ يمكن تغيير المتحول الأصم للمكمي الشمولي ∀ دون أن تتأثر دلالة الصيغة.

التوحيد

إذا حوت عبارتان حرْفيَّيْن متطابقين ومتنامين (أحدهما نفي الآخر)، فيمكن أن نحلهما بطريقة حساب

الفرضيات نفسها. وإذا كان لدينا حرفي $\lambda(\xi)$ في عبارة (حيث ξ متغير) وحرفي متمم $\neg\lambda(\tau)$ في عبارة أخرى (حيث τ حد لا يحوي المتغير ξ) فيمكن التعويض عن ξ بـ τ في العبارة الأولى، ثم القيام بحل الفرضيات على الحرفيات المتتامة للوصول لناتج حل **Resolvent** العبارتين.

ليكن لدينا مثلاً العبارتان **P(f(y),A)∨Q(B,C)** و**¬P(x,A)∨R(x,C)∨S(A,B)**. وبالتعويض عن **x** بـ **f(y)** في العبارة الثانية نحصل على **¬P(f(y),A)∨R(f(y),C)∨S(A,B)**. نلاحظ أن العنصرين في أول كل عبارة متتامان تماماً، ومن ثم يمكن الحل على هذا الحرفي **P(f(y),A)** لنحصل على ناتج الحل **R(f(y),C)∨S(A,B)∨Q(B,C)**.

يجري حساب التعويضات المناسبة باستخدام إجرائية تدعى التوحيد **unification**. وهي هامة جداً للذكاء الصنعي.

يمكن لحدود تعبير أن تكون متغيرات أو ثوابت غرضية أو تعابير وظيفية. نحصل على منتَسخ التعويض **substitution instance** لتعبير ما، بالتعويض عن متغيرات التعبير بحدود. لذا فإن للتعبير **P[x,f(y),B]** منتسخات التعويض التالية:

P[z,f(w),B]　　　**P[x,f(A),B]**　　　**P[g(z),f(A),B]**　　　**P[C,f(A),B]**

يدعى أول منتَسخ من اليسار بمتغير ألفبائي **alphabetic variant** لحرفي أصلي، لأننا لم نفعل سوى التعويض عن المتغيرات التي تظهر في التعبير **P[x,f(y),B]** بمتغيرات أخرى. يدعى آخر منتَسخ بالمنتسخ الأساسي **ground instance** إذ لا تحوي حدوده أي متغير.

يمكن تمثيل أي تعويض بمجموعة مرتبة من الثنائيات $s=\{\tau_1/\xi_1, \tau_2/\xi_2, ..., \tau_n/\xi_n\}$ حيث τ_i/ξ_i يعني أن كل ظهور للمتغير ξ_i سيعوض عنه بالحد τ_i على مدى التعويض. لاحظ أنه لا يمكن التعويض عن متغير بحد يحوي هذا المتغير. في التعبير السابق استعملنا التعويضات التالية للحصول على المنتسخات الأربعة:

التعبير	P[x,f(y),B]			
منتسخات التعويض	P[C,f(A),B]	P[g(z),f(A),B]	P[x,f(A),B]	P[z,f(w),B]
التعويضات المستخدمة	s4 ={C/x,A/y}	s3 ={g(z)/x,A/y}	s2 ={A/y}	s1 ={z/x,w/y}

يمكن تركيب تعويضين **s1** و**s2** وندون ذلك **s1s2**. للقيام بذلك نطبق أولاً **s2** على حدود **s1** ثم نضيف أي زوج من **s2** يحوي متغيرات غير موجودة في متغيرات **s1**. يمكن بيان أن تطبيق **s1** و**s2** بالترتيب على أي تعبير ω هو تطبيق **s1s2** نفسه على ω. أي $(\omega s1)s2=\omega(s1s2)$. كذلك يمكن بيان أن تركيب التعويضات تجميعي، أي **(s1s2)s3=s1(s2s3)**.

مثال

ليكن لدينا الصيغة ω التي هي **P(x,y)**، والتعويضان **s1={f(y)/x}** و**s2={A/y}** فيكون:

ω(s1s2)=[P(x,y)]{f(A)/x,A/y}=P(f(A),A) و (ωs1)s2=[P(f(y),y)]{A/y}=P(f(A),A)

لا تكون التعويضات عموماً تبديلية: أي لا يكون عموماً **s1s2=s2s1**. فترتيب تطبيق التعويضات يؤدي إلى اختلاف النتيجة.

مثال

باستخدام ω و**s1** و**s2** من المثال السابق

ω(s1s2)=P(f(A),A)
ω(s2s1)=[P(x,y)]{A/y,f(y)/x}=P(f(y) ,A)

عندما يطبق التعويض **s** على كل عنصر في مجموعة من التعابير {ω_i} ندون مجموعة منتسخات التعويض بِـ **{ω_i}s**. نقول إن مجموعة {ω_i} من التعابير قابلة للتوحيد **unifiable** إذا وجد تعويض **s** بحيث يكون **ω_1s = ω_2s = ω_3s =...** ونقول في هذه الحالة إن **s** هو الموحد **unifier** لِـ {ω_i} ذلك أنه يضم مجموعة الصيغ في أحادية **singleton**. على سبيل المثال ، **s={A/x,B/y}** يوحد المجموعة **{P[x,f(y),B],P[x,f(B),B]}** ليعطي **{P[A,f(B),B]}**. ومع أن **s={A/x,B/y}** هو موحد للمجموعة**{P[x,f(y),B],P[x,f(B),B]}** فإنه ليس الموحد الأبسط، إذ نلاحظ أننا لسنا مضطرين لإبدال (للتعويض عن) **x** بِـ **A** لبلوغ التوحيد. نقول إن موحداً **g** لِـ {ω_i} هو الموحد الأعم (أو الأبسط) **most general unifier (mgu)**، إذا كان أي موحد آخر **s** لِـ {ω_i} يعطي **{ω_i}s** فحينئذ يوجد تعويض آخر **'s** بحيث **{ω_i}s={ω_i}gs'**. فضلاً على ذلك، المنتسخ المشترك الناتج عن الموحد الأعم وحيد (ماعدا اختلافات أبجدية).

يوجد العديد من الخوارزميات المستخدمة لإيجاد الموحد الأعم لمجموعة منتهية من التعابير القابلة للتوحيد (وهي تعلن الإخفاق إذا كانت التعابير غير قابلة للتوحيد).

الخوارزمية **UNIFY** المستخدمة هنا، مأخوذة عن أحد المراجع. وهي تعمل على تعابير موضوعة في قائمة مهيكلة **list-structured**، وكل حرفي وكل حد فيها مكتوب كقائمة. فمثلاً الحرفي **¬P(x,f(A,y))** يكتب بالشكل المهيكل **(¬Px(f A y))**، ويكون التعبير **¬P** هو التعبير الأول من المستوى الأعلى والتعبير **(f A x)** هو التعبير الثالث من المستوى الأعلى.

نستخدم في الخوارزمية **UNIFY** المفهوم الأساسي وهو مجموعات عدم التوافق **disagreement set**. نحصل على مجموعة عدم التوافق لمجموعة غير خالية ***W*** من التعابير بتحديد موقع أول رمز (بدءاً من اليسار) لا تكون عنده رموز كل التعابير في ***W*** متطابقة، ثم نستخرج من كل تعبير في ***W*** التعابير الجزئية التي تبدأ بالرمز الذي يحتل هذا الموقع. تكون مجموعةُ هذه التعابير الجزئية المتتالية مجموعةَ عدم التوافق لِـ ***W***. على سبيل المثال، إذا كانت **{(¬Px(f A y)) , (¬Px(f z B))}** هي مجموعة قائمتين

فإن مجموعة عدم التوافق لها هي **{A,z}**. يمكن جعل مجموعة عدم التوافق متوافقة باستخدام التعويض **A/z**.

إن الخوارزمية **UNIFY** إما توجد الموحد الأعم لمجموعة تعابير قابلة للتوحيد، وإما تعلن الإخفاق إذا كانت التعابير غير قابلة للتوحيد. ونتيجة الخوارزمية هي الموحد الأعم وهو زوج من تعابير القوائم المهيكلة.

مثال: التعويض **s={A/x,B/y}** يوحد المجموعة:**{P[A,f(y),B],P[x,f(B),B]}** ليعطي **{P[A,f(B),B]}**.

	UNIFY (Γ)	حيث Γ مجموعة قوائم مهيكلة من التعابير
1	**k ← 0; Γ_k ← Γ; σ_k ← ε**	الخطوة الابتدائية. (ε هو التعويض الفارغ)
2	**if Γ_k is singleton, exit with σ_k, the mgu of Γ, otherwise continue**	إذا كانت Γ_k أحادية، أُخرُجْ من الإجراء مرجعاً σ_k وهي الموحد الأعم لِ Γ، وإلا فتابع
3	**D_k ← the disagreement set of Γ_k**	نضع في **D_k** مجموعة عدم التوافق لِ Γ_k
4	**If there exits elements v_k, t_k in D_k such that v_k is a variable that does not occur in t_k continue. Otherwise exit with failure; Γ is not unifiable**	إذا وجد عنصران **v_k** و**t_k** في المجموعة **D_k** حيث **v_k** متغير لا يظهر في **t_k** تابع، وإلا فاخرُج وأعلِن الإخفاق، وتكون Γ غير قابلة للتوحيد
5	**σ_{k+1} ← σ_k{t_k/v_k}; Γ_{k+1} ← Γ_k{t_k/v_k}**	لاحظ أن **$\Gamma_{k+1} = \Gamma_k\sigma_{k+1}$**
6	**k ← k+1**	أضِف واحداً إلى **k**
7	**Go to 2**	اذهب إلى الخطوة **2**

الحل

لتطبيق الحل هنا، بهدف المحاكمة على المعرفة نحتاج أولاً إلى تحويل الصيغ الجيدة التركيب **wff** إلى عبارات ويتم ذلك باتباع الخطوات التالية:

- حذف الاقتضاء (عبارة مكافئة مع ∨، مثلاً نستعيض عن **x ⇒ y** بِ **¬x ∨ y**)
- تقليص مدى النفي (تطبيق قوانين دومورغان)
- تقييس المتغيرات (لكل مكمي وجودي رمز خاص به)، مثلاً يمكن كتابة الصيغة **(∀x)[¬P(x)∨(∃x)Q(x)]** هكذا **(∀x)[¬P(x)∨(∃y)Q(y)]**.
- حذف المكميات الوجودية (باستخدام دوال سكوليم)، **مثال: (∀x)(∃y) Height(x,y)** تصبح **(∀x) Height(x,h(x))**.
- التحويل إلى الشكل الأمامي **Prenex Form** (وضع جميع المكميات في المقدمة)
- وضع المصفوفة في شكل العطف النظامي
- حذف المكميات الشمولية
- حذف رمز العطف ∧

➢ إعادة تسمية المتغيرات بحيث لا يظهر المتغير في أكثر من عبارة

بعد إجراء هذه التحويلات نستخدم الحل بالنقض كما في حالة حساب الفرضيات.

الحل

لنفترض أن الربوط يعرف أن جميع الطرود الموجودة في الغرفة **27** أصغر من أي طرد موجود في الغرفة **28**. أي:

$$\forall(x,y)\{package(x)\wedge package(y)\wedge inroom(x,27)\wedge inroom(y,28)\}\supset smaller(x,y)$$

تصبح

$$\neg P(x)\vee\neg P(y)\vee\neg I(x,27)\vee\neg I(y,28)\vee S(x,y)$$

حيث: **P = package, I = inroom, S = smaller**

فإذا كان لدى الربوط الحقائق التالية:

P(A),P(B) و **I(A,28)** $\vee$**I(A,27)** و **I(B,27)** و $\neg$**S(B,A)**

أي **A** و**B** أغراض و**A** موجود في الغرفة **27** أو الغرفة **28** والغرض **B** في الغرفة **27** و**B** ليس أصغر من **A**.

المطلوب: إثبات أن **A** في الغرفة **27** بطريقة الحل بالنقض.

تبين الشجرة طريقة الحل هذه.

¬I(A,27)
نفي الصيغة المطلوب إثباتها
I(A,27)∨ I(A,28)
I(A,28)
¬P(x)∨ ¬P(y)∨ ¬I(x,27) ∨ ¬I(y,28) ∨ S(x,y)
¬P(x)∨ ¬P(A)∨ ¬I(x,27) ∨ S(x,A)
P(A)
¬P(x)∨ ¬I(x,27) ∨ S(x,A)
P(B)
¬I(B,27)∨ S(B,A)
I(B,27)
S(B,A)
¬S(B,A)
Nil

الحل بالنقض

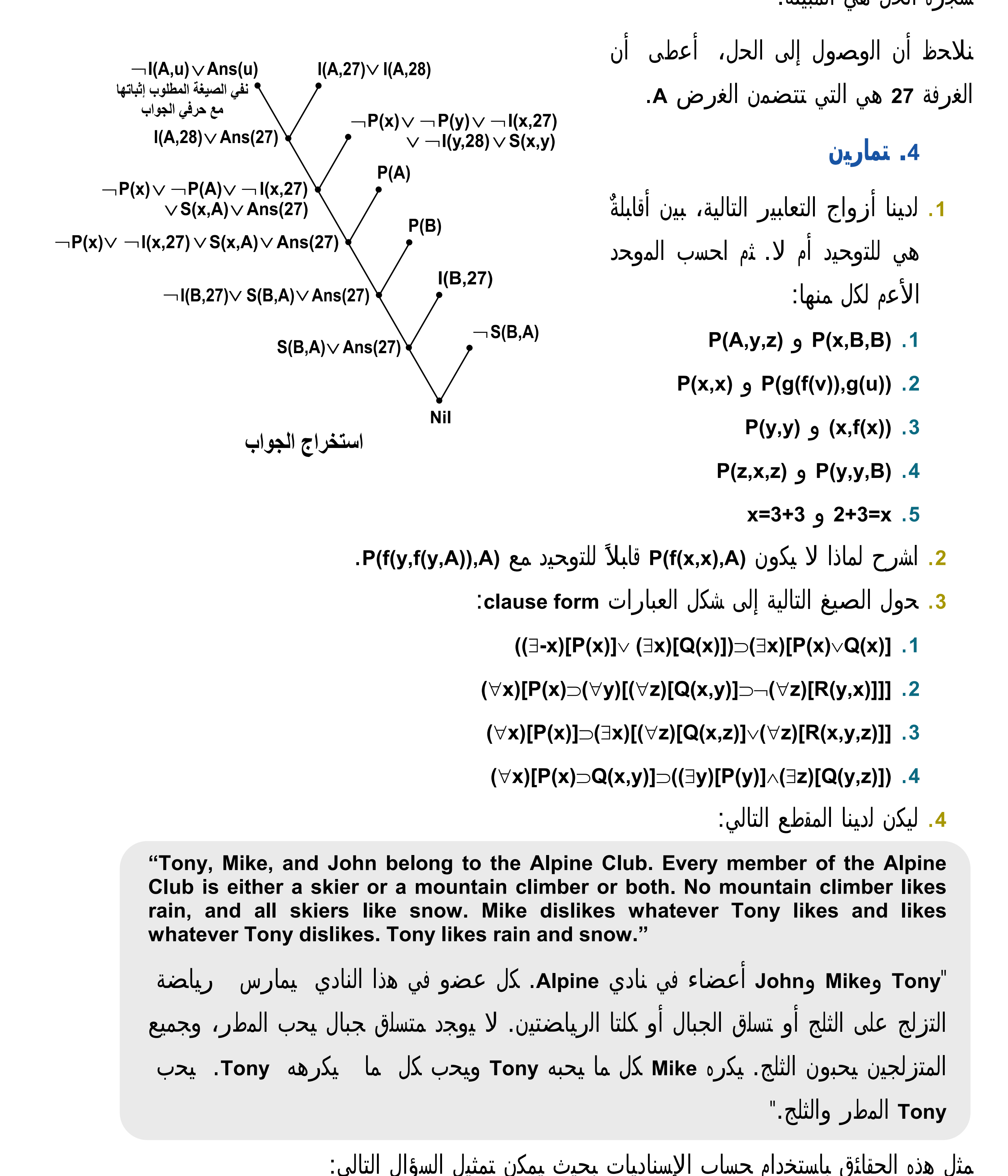
استخراج الجواب

قد نتطلب إيجاد الحل. أي لا نثبت الحل بنقض الفرض وإنما نبحث عنه. نريد إيجاد كل التعويضات التي أدت إلى الوصول إلى الحل (أي إعطاء تبرير لهذا الحل).

في المثال السابق نود أن نجيب على السؤال (في أي غرفة يوجد الطرد **A**). لذلك ، نضع **Ans (u)** ونريد أن نبرهن على: **I(A,u) (∃u)** أي إذا كان الجواب **u**، نود أن نوجد الغرفة **u** التي فيها الغرض **A**.

شجرة الحل هي المبينة.

نلاحظ أن الوصول إلى الحل، أعطى أن الغرفة **27** هي التي تتضمن الغرض **A**.



استخراج الجواب

4. تمارين

1. لدينا أزواج التعابير التالية، بين أقابلةٌ هي للتوحيد أم لا. ثم احسب الموحد الأعم لكل منها:

1. **P(x,B,B)** و **P(A,y,z)**
2. **P(g(f(v)),g(u))** و **P(x,x)**
3. **(x,f(x))** و **P(y,y)**
4. **P(y,y,B)** و **P(z,x,z)**
5. **2+3=x** و **x=3+3**

2. اشرح لماذا لا يكون **P(f(x,x),A)** قابلاً للتوحيد مع **P(f(y,f(y,A)),A)**.

3. حول الصيغ التالية إلى شكل العبارات **clause form**:

1. $((\exists\text{-}x)[P(x)]\vee(\exists x)[Q(x)])\supset(\exists x)[P(x)\vee Q(x)]$
2. $(\forall x)[P(x)\supset(\forall y)[(\forall z)[Q(x,y)]\supset\neg(\forall z)[R(y,x)]]]$
3. $(\forall x)[P(x)]\supset(\exists x)[(\forall z)[Q(x,z)]\vee(\forall z)[R(x,y,z)]]$
4. $(\forall x)[P(x)\supset Q(x,y)]\supset((\exists y)[P(y)]\wedge(\exists z)[Q(y,z)])$

4. ليكن لدينا المقطع التالي:

> **"Tony, Mike, and John belong to the Alpine Club. Every member of the Alpine Club is either a skier or a mountain climber or both. No mountain climber likes rain, and all skiers like snow. Mike dislikes whatever Tony likes and likes whatever Tony dislikes. Tony likes rain and snow."**
>
> "**Tony** و**Mike** و**John** أعضاء في نادي **Alpine**. كل عضو في هذا النادي يمارس رياضة التزلج على الثلج أو تسلق الجبال أو كلتا الرياضتين. لا يوجد متسلق جبال يحب المطر، وجميع المتزلجين يحبون الثلج. يكره **Mike** كل ما يحبه **Tony** ويحب كل ما يكرهه **Tony**. يحب **Tony** المطر والثلج."

مثل هذه الحقائق باستخدام حساب الإسناديات بحيث يمكن تمثيل السؤال التالي:

"Who is a member of the Alpine Club who is a mountain climber but not a skier?"

"من هو العضو في النادي الذي يمارس التسلق إلا أنه لا يتزلج؟"

كتعبير في الحساب الإسنادي. استخدم الحل بالنقض مع استخراج الجواب للإجابة عن هذا السؤال.

5. لنفترض أننا نعرف الحقائق التالية عن الفيلة الثلاثة Sam وClyde وOscar:

1. **Sam is pink.**
2. **Clyde is gray and likes Oscar.**
3. **Oscar is either pink or gray (but not both) and likes Sam.**

1. لون سام **Sam** وردي.
2. لون كلايد **Clyde** رمادي وهو يحب **Oscar**.
3. لون أوسكار **Oscar** إما وردي وإما رمادي (ولكن ليس كليهما)، وهو يحب **Sam**.

استخدم الحل بالنقض لإثبات أنه يوجد فيل رمادي يحب فيلاً وردياً. أي أثبت أنه:

$$(\exists x,y)[Gray(x) \wedge Pink(y) \wedge Likes(x,y)]$$

5. مراجع البحث

- كتاب "الذكاء الصنعي رؤية جديدة" من مطبوعات الجمعية العلمية السورية للمعلوماتية **2004**
- **Giarratano & Riley "Expert Systems: Principles and Programming", 3d Edition, 1998**

الفصل الثالث: طرق أخرى لتمثيل المعرفة والنظم الخبيرة

1. نظم قواعد الإنتاج

عرفنا سابقاً عبارات هورن التي تحوي حرفياً موجباً على الأكثر. ورأينا أن لدينا:

- قواعد (علاقات اقتضاء)
- حقائق
- أهداف نود تحقيقها (برهنتها)، بناء على القواعد والحقائق المتوفرة.

تسمح لغة **Prolog** ببرمجة مثل هذه النظم بسهولة.
يمكن أيضاً استخدام لغة **Clips** لتحقيق ذلك.

يوجد العديد من النظم الخبيرة التي تعتمد هذا الأسلوب:

لدينا قاعدة المعرفة التي تتضمن الحقائق (في ذاكرة العمل)، والقواعد في قاعدة المعرفة، ولدينا الأهداف (في ذاكرة العمل)، ولدينا محرك استدلال يسعى إلى إجراء محاكمة لمعرفة صلاحية الأهداف.

يمكن أن يعمل محرك الاستدلال بالسلسلة الأمامية **Forward chainig** انطلاقاً من الحقائق الموجودة ، أو بالسلسلة الخلفية **Backward Chaining** انطلاقاً من الهدف.

عبارات هورن

هي العبارات التي تحتوي حرفياً موجباً واحداً على الأكثر ولها ثلاثة أنواع:

- العبارة الوحيدة التي تحوي ذرة وحيدة وندعوها حقيقة **Fact**. (مثل **P** و**Q**)
- الاقتضاء (و الذي ندعوه عادةً قاعدة **Rule**). تتألف مقدمته من عطف حرفيات موجبة ونتيجته من حرفي موجب وحيد. (مثل $P \wedge Q \supset R$)
- مجموعة من الحرفيات السالبة والمكتوبة على شكل اقتضاء مقدمته تتألف من عطف حرفيات موجبة ونتيجته المجموعة الخالية. يمكن الحصول على هذا الشكل مثلاً عند نفي صيغة **wff** مطلوب إثباتها تتألف من عطف حرفيات موجبة. ندعو هذه الصيغة عادةً بالهدف **Goal**. (مثل $P \wedge Q \supset$)

السلسلة الأمامية

انطلاقاً من الحقائق الموجودة في ذاكرة العمل، يجب البحث عن القواعد التي يمكن تطبيقها واستنتاج حقائق جديدة.

مثال: إذا كانت لدينا القواعد التالية:

إذا كان **x** فيل فإن **x** من الثديات: $elephant(x) \supset mammel(x)$

إذا كان x من الثديات فإن x حيوان: **mammel(x)⊃animal(x)**

والحقيقة:

Clide هو فيل: **elephant(clide)**

بالسلسلة الأمامية نرى أنه يمكن تطبيق القاعدة الأولى وبتوحيد **x** مع **clide** نستنتج أن **mammel(clide)** أي إن **clide** من الثديات.

ثم بتطبيق السلسلة الأمامية ثانية وتطبيق القاعدة الثانية ، وتوحيد **x** مع **clide** نستنتج أن **animal(clide)** أي إن **clide** حيوان. وبذلك يضاف إلى ذاكرة العمل الحقيقتان : **mammel(clide)** و **animal(clide)**.

تُستخدم السلسلة الأمامية في تشخيص الأمراض، حيث إن تأكيد وجود مرض يضيف حقائق أعراض المرض إلى ذاكرة العمل.

السلسلة الخلفية

انطلاقاً من الهدف الذي نود تحقيقه، يجب البحث عن الفرضيات التي تدعم تحقيق هذا الهدف.

ففي المثال السابق إذا اعتبرنا القواعد نفسها موجودة وهي:

إذا كان x فيل فإن x من الثديات: **elephant(x)⊃mammel(x)**

إذا كان x من الثديات فإن x حيوان: **mammel(x)⊃animal(x)**

وهدفنا أن نبرهن أن **clide** حيوان أي: **animal(clide)**

نبحث عن قاعدة نتيجتها الوصول إلى **animal(clide)** فنجد أن القاعدة الثانية تؤدي إلى استنتاج هذا ، إذا تحقق لدينا **mammel(clide)** فيصبح الهدف الجديد الآن البحث عن فرضيات لبرهان الهدف الجديد **mammel(clide)**. نجد أنه بتطبيق القاعدة الأولى نصل إلى هذا الهدف إذا كان لدينا **elephant(clide)**. وبما أن هذه الحقيقة موجودة في ذاكرة العمل، فإن فرضية البدء صحيحة أي **animal(clide)** صحيحة.

مقارنة بين السلسلة الأمامية والسلسلة الخلفية

	السلسلة الأمامية	السلسلة الخلفية
الاستخدام	التحكم، التخطيط	التشخيص
مقاد بِ	المعطيات	الهدف
المحاكمة	من الأسفل إلى الأعلى، إيجاد النتائج التي تدعمها المعطيات	من الأعلى إلى الأسفل، إيجاد حقائق تدعم فرضيات
يشبه خوارزميات البحث	عرضاً أولاً	عمقاً أولاً

مشاكل قواعد الإنتاج وحلول

- غالباً لا يكون التمثيل البسيط فعالاً.
- لا يمكن التعبير عن بعض المعارف ببساطة باستخدام هذه القواعد.
- يصعب فهم وصيانة هذه القواعد حين تغدو ذات عدد ضخم جداً.

في البحث عن طرق تمثيل أخرى. نحتاج إلى تمثيل الأغراض المعقدة في العالم وخواصها وعلاقاتها بالأغراض الأخرى، فنبحث عن:

- تمثيل جيد مناسب لجميع المعارف في النطاق مهما تعقدت.
- إمكان استنتاج معارف أخرى من التمثيل.
- أن يحمل التمثيل ما يساعد على جعل الاستدلال فعالاً (التركيز باتجاه المعرفة الضرورية).
- تحصيل المعرفة على نحو فعال.

سنحتاج إلى معارف أكثر شمولاً لإنجاز نظم ذكاء صنعي، متعددة البراعات، من المستوى البشري . ستحتاج مثل هذه النظم إلى معارف عن العديد من المواضيع التي برهن على صعوبة نمذجتها صورياً. وربما كان من المتناقضات أن علماء الذكاء الصنعي يجدون تمثيلَ المواضيع، السهلة الفهم على طفل في العاشرة من عمره، أشد مقاومةً لطرائق الذكاء الصنعي من تمثيل المواضيع التي أمضى الخبراء سنوات في تحصيلها. إذ يستطيع الفيزيائيون إعطاء وصف دقيق ومفَصل لظواهر فيزيائية باستخدام المعادلات الموجية ونظرية النسبية وبنى رياضية أخرى، أما الباحثون في الذكاء الصنعي فمازالوا يتجادلون حول أفضل طريقة لتمثيل حقائق بسيطة (إلا أنها جد مفيدة) مثل كون السائل يطابق شكل الفنجان الذي نضعه فيه، وأنه ينسكب إذا قلبنا الفنجان. ويبدو أن المميزات المغرقة في النظرية، التي ابتكرها الفيزيائيون والرياضيون أسهل صوغاً من الأفكار التي مكنت البشر برغم كل شيء، من أداء وظائفهم أداء جيداً تقريباً، حتى قبل ظهور أرسطو.

قد يكون أحد أسباب صعوبة تمثيل المعارف البديهية هو مجرد ضخامة هذه المعارف. يمكن تقسيم الكثير من المعارف الخبيرة بحيث تكفي المئات أو الآلاف من الحقائق لبناء نظم خبيرة مفيدة . كم سنحتاج من الحقائق لبناء نظام يصل إلى مستوى الذكاء البشري العام؟ لا أحد يعرف على وجه اليقين! يعتقد دوغ لينا **Doug Lenat** الذي يقود جهوداً لبناء قاعدة معرفة واسعة لمثل هذه الحقائق ، اسمها **CYC**، أننا سنحتاج إلى ما بين مليون وعشرة ملايين من هذه الحقائق.

تكمن مشاكل أخرى لصياغة مثل هذه المعارف في أنه ليس من السهل التعبير عن بعض المواضيع بجمل تصريحية. على سبيل المثال، ليس من السهل وصف الأشكال والأجسام الفيزيائية المعقدة باستعمال الجمل. فمثلاً، هل يمكن وصف وجه إنسان بكلمات، بحيث يمكن أن يتعرفه شخص آخر لم

يره قبلاً؟ كيف يمكن أن نستخدم الكلمات لوصف شجرة أو منظر جبلي أو غروب شمس مداري؟ وإذا كنا غير قادرين على وصف شيء باستخدام الإنكليزية أو أية لغة طبيعية أخرى، فهذا سبب كاف لنعتقد أننا لن نجد نمذجة لهذه المفاهيم بلغة المنطق أيضاً.

2. شبكات الدلالة

لا يختلف إلا القليلون حول قيمة "ربوط منزلي". هذه الآلة التي يمكن أن يعهد إليها بمهام تنظيف المنزل وغسل الثياب وتحضير الوجبات وتقديمها، وإنجاز إجراءات الصيانة المنزلية (مثل تبديل مصابيح الإضاءة المحترقة)، وغسل الأطباق، وغيرها. ولكن، فكر في كمية المعارف التي سيحتاج إليها هذا الربوط! سيحتاج إلى معارف كافية لتمكّنه من الإجابة عن أسئلة مثل:

أين تَضع مصابيح الإضاءة المحترقة؟
كيف تُمسك بصحن؟ أو بكأس؟
أين تضع جميع الأشياء التي تُخرجها من غسالة الصحون؟
كيف تُخبرنا بأن المكنسة الكهربائية بحاجة إلى تفريغ؟
كم من الوقت يمكن أن يبقى الخس، عادة، في الثلاجة؟

وغيرها وغيرها.

يمكن القول أيضاً أن المعارف البديهية هي التي تجعل النظام الخبير أكثر فائدة. ذلك أن النظم الخبيرة تنجز عملها جيداً في مجالات خبرتها فقط. إلا أن إضافة المعارف البديهية إليها يجعلها قادرةً ، على الأقل، على أن تدرك المطلوب حين يطلب المستخدم معلومات خارج تلك المجالات. إضافة إلى ذلك ، تمكّن هذه المعارف النظام من أن يتنبأ بدقة:

- متى تكون المعرفة لها صلة بالمهمة التي ينفذها الربوط؟
- ومتى لا تكون كذلك؟

يمكن، في كثير من الأحيان، ترتيب كيانات المعارف البديهية ومعارف مجالات الخبرة في بنى تراتبية تنظم عملية المحاكمة وتسهلها. ففي تمثيل المعارف البديهية في النظام **CYC**، على سبيل المثال، يشار إلى الكيان الأساسي جداً بالثابت الغرضي **Thing** (شيء). يوجد في **CYC** عدة أنواع من الأشياء **things**: أغراض في العالم، الأغراض الرياضية، الأحداث والإجرائيات وغيرها. تُرتب هذه الأشياء في تصنيف (أو تراتب) يرمز ضمنياً حقائق من الشكل:

" X is a P, all P's are Q's, All Q's are R's"

"**X** هي من النوع **P**، كل عناصر **P** هي عناصر من **Q**، و كل عناصر **Q** هي عناصر من **R**" وغيرها.

يمكن تمثيل تراميز التراتبات التصنيفية باستخدام الشبكات أو باستخدام بنى معطيات تُدعى الأطر **frames**. سنناقش أولاً التمثيل باستخدام الشبكات باستعمال معلومات عن الآلات المكتبية لتكون مثالاً توضيحياً.

لنفترض أننا نريد مثلاً تمثيل الحقائق التالية: **Snoopy** هي طابعة ليزرية، كل الطابعات الليزرية هي طابعات، كل الطابعات هي آلات، إضافةً إلى بعض المعلومات الأخرى المتعلقة بهذه المسألة. يمكن أن يكون لدينا عبارات الحساب الإسنادي التالية:

Laser_Printer(Snoopy)
(∀x) [Laser_Priner(x) ⊃ Printer(x)]
(∀x) [Priner(x) ⊃ Office_Machine(x)]

تُمثل كل من **Laser_Printer** و **Printer** و **Office_Machine** فئات **categories** في التراتبية.

- ينطوي أحد أنواع المحاكمة المهمة التي تستعمل المعرفة التصنيفية على تعدي الفئات . فمثلاً يمكننا، من الحقائق والقواعد السابقة، استنتاج ما يلي:

Office_Machine(Snoopy) و **(∀x) [Laser_Priner(x) ⊃ Office_Machine(x)]**

- يمكن أن يكون لجميع عناصر الفئات التصنيفية خواص معينة. فمثلاً مصدر الطاقة لجميع آلات المكتب هو مأخذ كهربائي في الجدار. يمكن التعبير عن هذه الخواص باستخدام دالّة وإسنادية المساواة:

(∀x)[Office_Machine(x) ⊃ [Energy_Source(x) = Wall_Outlet]]

- لاحظ أن عناصر الفئات الجزئية ترث نموذجياً الخواص المشتركة لفئاتها العليا:

(∀x)[Laser_Printer(x) ⊃ [Energy_Source(x) = Wall_Outlet]]

تُسهل هذه الأنواع العامة من الاستدلالات المتعلقة بالكيانات في التراتبية التصنيفية، باستعمال تمثيل بياني يسمى شبكة الدلالة **Semantic network**.

تعريف:

شبكات الدلالة هي بنى بيانية، تُستخدم لترميز المعارف التصنيفية عن الأغراض وخواصها.

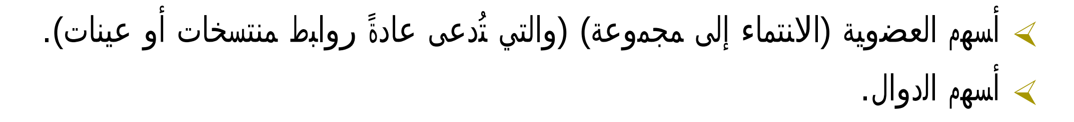
سنُناقش فيما يلي نسخة مبسطة منها وذلك بهدف إظهار الأفكار الرئيسية.

يوجد نوعان من العقد:

- العقد المعنونة بثوابت علاقاتية وهي تُقابل فئات تصنيفية أو خواص.
- العقد المعنونة بثوابت غرضية تُقابل أغراضاً في المجال.

كما يوجد ثلاثة أنواع من الأسهم تربط هذه العقد:

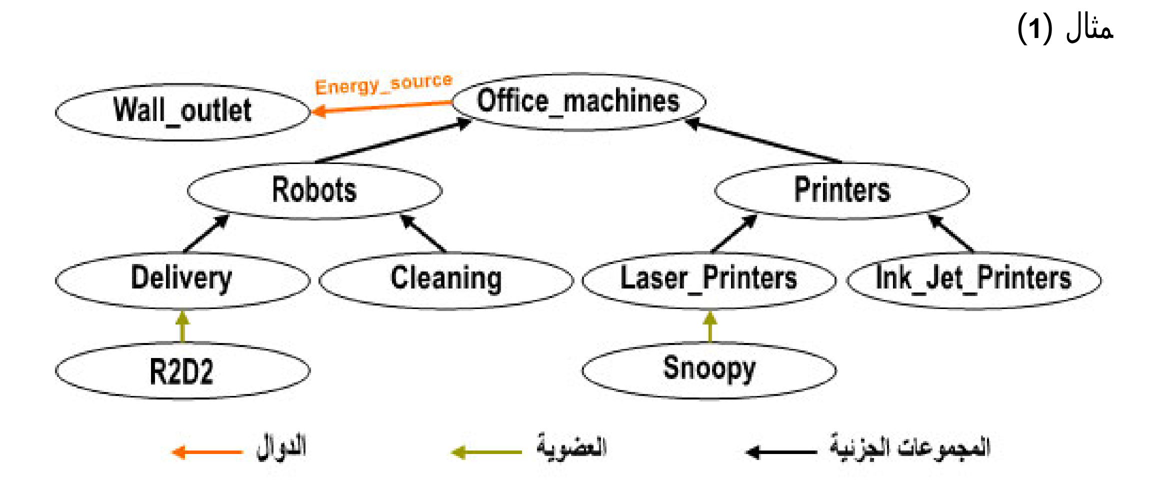
- أسهم المجموعات الجزئية (والتي تُدعى عادةً **isa links**، أي روابط من النوع **isa**)



- أسهم العضوية (الانتماء إلى مجموعة) (والتي تُدعى عادةً روابط منتسخات أو عينات).
- أسهم الدوال.

مثال على شبكة دلالة.

مثال (1)

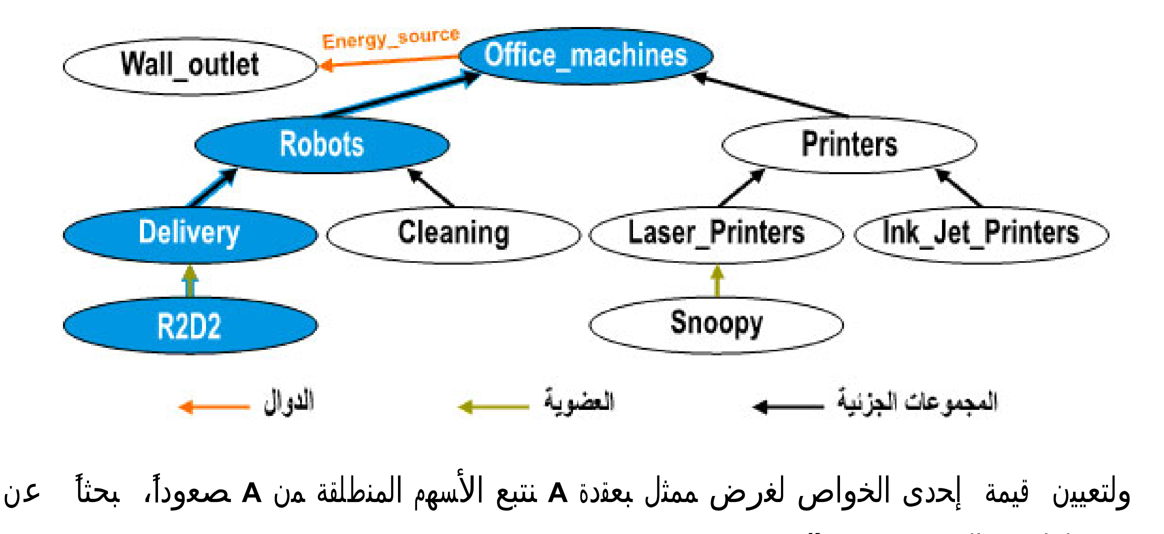


مثال (2)

إن المحاكمة المتعلقة بالخواص أو الانتماء إلى مجموعة أسهل وأشد فاعلية باستخدام الشبكات الدلالية من المحاكمة باستخدام الحل غير الموجه على الصيغ الجيدة التركيب الموافقة.

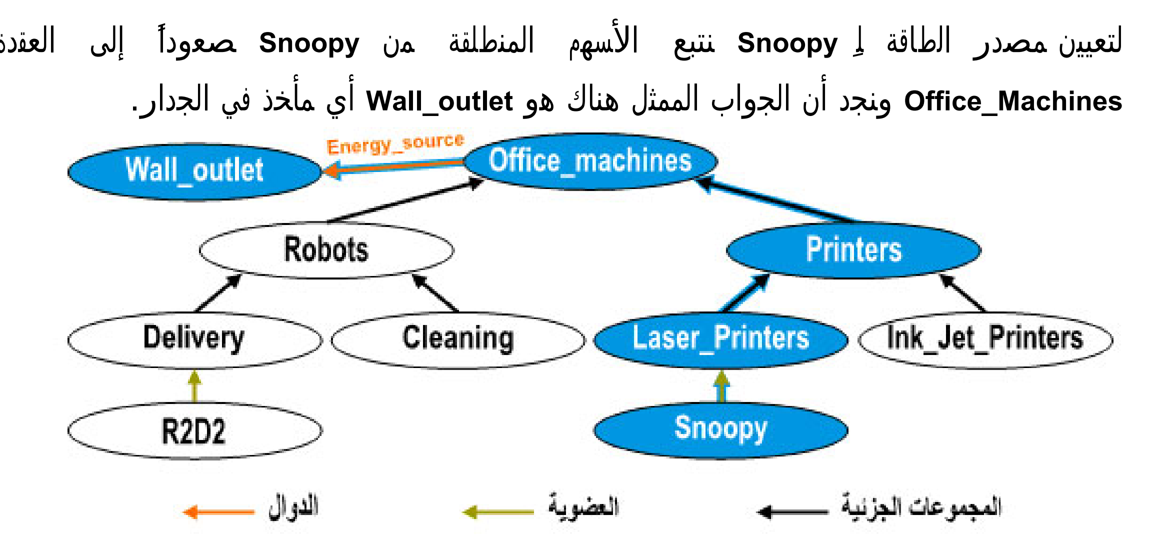
لكي نصل إلى قرار بشأن غرضٍ ممثلٍ بعقدة **A**: أهو عضو من مجموعة ممثلة بعقدة **B**، يجب اتباع الأسهم صعوداً من **A** لنرى: هل سنقابل العقدة **B** (سنصل إلى **B**). **مثال**

يمكننا بسهولة، أن نقرر أن الآلة **R2D2** هي آلة مكتبية **Office_Machine** على سبيل المثال.



ولتعيين قيمة إحدى الخواص لغرض ممثل بعقدة **A** نتبع الأسهم المنطلقة من **A** صعوداً، بحثاً عن عقدة لها هذه الخصيصة. **مثال**

لتعيين مصدر الطاقة لِ **Snoopy** نتبع الأسهم المنطلقة من **Snoopy** صعوداً إلى العقدة **Office_Machines** ونجد أن الجواب الممثل هناك هو **Wall_outlet** أي مأخذ في الجدار.



إننا نقوم في غالب الأحيان باستدلالات مفتَقَلَة **by default**؛ وهي استدلالات تحاول حذف المعرفة التي تؤدي إلى نقيض ما نرغب في افتراضه صحيحاً. ولكن إذا وصلت معارف جديدة مناقضة، فعلينا أن نقوم بسحب الاستدلال المفتَقَل. لقد اقتُرِح العديد من النظم والآليات المنطقية للإمساك بهذه الظاهرة ، أبسطها إلغاء التوريث.

لنفترض، في المثال السابق، أننا نريد القول بأن مصدر الطاقة المفتقَل للآلات المكتبية ، هو مأخذ كهربائي في الجدار. ولكن يوجد استثناء لهذه القاعدة، وهو أن مصدر طاقة الربوط هو البطارية. يمكن التعبير عن هذه المعرفة بإضافة سهم وظيفي آخر إلى الشبكة الدلالية كما هو مبين في الشكل.

قد يبدو أن آلية وراثة الخواص ستقود إلى تناقض في الشبكة الجديدة. مع أن الوراثة تسمح لنا باستنتاج (وهذا صحيح) أن مصدر طاقة الطابعات هو مأخذ كهربائي جداري، فإنها تُناقض أيضاً الحقيقةَ التصريحية الممثلَة بحقيقة أن مصدر طاقة الربوط هو البطارية (بافتراض أن البطارية والمأخذ الجداري مختلفان). إننا محميون من الوقوع في هذا التناقض بحسب الطريقة التي تُستَعمل فيها الشبكة. فللمعلومات عن أكثر الأصناف خصوصية (مرتبة حسب أقواس المجموعات الجزئية أو أقواس المنتسخات) أفضليةٌ على الأصناف التي هي أقل خصوصية. لذلك، فإننا في الحالة التي نحاول فيها الاستفسار عن مصدر طاقة **R2D2** نبحث أولاً عن موضع العقدة المعنونة **R2D2** في الشبكة، ثم ننظر: هل للعقدة **R2D2** سهم دلالي معنون بمصدر الطاقة؟ ففي حالة الإيجاب، نقبل الجواب المعطى بالعقدة الموجودة في رأس هذه القوس كعقدة نهائية. وإلا، فإننا سنتبع الأسهم بصعود تراتبي إلى أن نصل إلى أول عقدة لها سهم دلالي معنون بمصدر الطاقة، ثم نستعمل العقدة على رأس هذا السهم للجواب عن هذا الاستفسار.

إن المعلومات المرافقة للعقد العليا في التراتب التصنيفي (الهرمية) هي عادةً معلومات مفتَقَلَة عامة ، مرشحة للإلغاء من قبل معلومات أكثر خصوصية مرافقة لعقد أدنى في التراتبية.

فكما يظهر الشكل، معلومات مصدر الطاقة **Energy_source** للعقدة العليا **Office_machines** والمتمثلة بالمأخذ الجداري **Wall_outlet** هي معلومات مفتَقَلَة عامة، تُلغى من قبل معلومات مصدر

الطاقة **Energy_source** الأكثر خصوصية للعقدة الأدنى **Robots** والمتمثلة بالبطارية **Battery**.

مواضيع تتعلق بالتنفيذ العملي

تمثيل شبكات الدلالة بسيط وفعال

- جداول تسرد الأغراض وخواصها
- جداول أو قوائم **lists** بالعلاقات

يمكن التحول منه إلى طرائق تمثيل أخرى

- في حساب الإسناديات:
  - العقد ثوابت أو متغيرات
  - الوصلات: إسناديات
- في حساب الفرضيات:
  - العقد والعلاقات هي فرضيات ترتبط بروابط منطقية مناسبة

الغرض	الواصفة	قيمة الواصفة
Astérix	المهنة	محارب
Obélix	الحجم	كبير جداً
Idéfix	الحجم	صغير
Panoramix	الحكمة	لا نهائية

يمكن إدراج المعارف كثلاثيات (غرض - واصفة - قيمة) OAV

- يمكن استخدام هذه الثلاثيات لتمثيل شبكات الدلالة
- تقود بسرعة إلى حجوم هائلة من المعلومات

مشاكل شبكات الدلالة

التعبيرية

- لا توجد بنى داخلية للعقد، فهي مجرد أغراض فقط.
- لا يوجد طريقة سهلة لتعريف التجريبيات (الكسبيات) **heuristics**.
- يمكن توسيعها لتعبر عن معارف أعقد وأعمق، ولكن هذا يؤدي إلى حجوم هائلة.
- ملائمة فقط للعلاقات الثنائية، وليس الثلاثية أو المتعددة.

الفعالية

- نصل بسرعة إلى عدد هائل من العقد والعلاقات.
- يؤدي البحث فيها إلى توسعات انفجارية خاصة إذا كانت نتيجة البحث سلبية.

استخدامها

- لا يوجد معايير لأنواع الوصلات، باستثناء وصلات التعريف **is-a** وربط الجزء بالكل **part-of** وربط الصفوف بصفوف آباء **a-kind-of** أو اختصاراً **AKO** ولا لتسميات العقد.
- تعتمد المعرفة المستنتجة على خاصية الوراثة، ولكن أحياناً توجد استثناءات في الصف الابن ، مثال: يتمتع صف الطيور بخاصة الطيران، صف الدجاج منه لا يطير

مثال: تحويل إلى شبكات دلالة

Human(Marc)	⟹	is-a(Marc, Human)
Score(Wahda, Itihad (75 73))	⟹	is-a(G23, game) local_team(G23, Wahdah) visitor_team(G23, Itihad) score(G23, 75_73)

ملاحظة: شبكات الدلالة مفيدة في وصف دلالة اللغات كما في المثال:

"Ahmad gave the book to Mona"

is_a(Action, give);
Tense(Action, past);
Subject(Action, Ahmad)
Obj1(Action, Mona)
Obj2(Action, book).

3. ترابط المفاهيم

ترابط المفاهيم: Conceptual Dependencies

هو طريقة للتعبير عن دلالة جمل باللغة الطبيعية بحيث:

- نسهل استنتاج الاستدلال من الجمل.
- يكون التمثيل مستقلاً عن اللغة التي قيلت فيها الجملة (عربية أم انكليزية...).

طُورت هذه الطريقة في السبعينيات بواسطة **Schank** ونفذت في العديد من البرمجيات لفهم اللغات الطبيعية.

مثال: Ahmad gave the book to Mona (أحمد أعطى الكتاب إلى منى)



في هذا المثال:

- الأسهم تشير إلى اتجاه الترابط.
- السهم المزدوج يشير إلى علاقة باتجاهين بين الفاعل والفعل.
- **P**: تعني الزمن الماضي للفعل.
- **ATRANS**: أحد الأساسيات **Primitives** ويشير إلى نقل ملكية.
- **O**: غرض **object**.
- **R**: المستقبل **receiver**.

في ورشة العمل بعنوان **Arabic Natural Language Processing** التي أقيمت في المغرب في نيسان **April 2007** على هامش مؤتمر **ictis'07** (انظر الموقع: **www.ictis07.org**) جرى التصريح عن

تعتمد المفاهيم Universal Networking Language (UNL) لتمثيل اللغة العربية، وأنه جرى الاتفاق بين ممثلين عن **15** لغة للتمثيل بهذه اللغة. المشروع مدعوم من هيئة الأمم المتحدة.

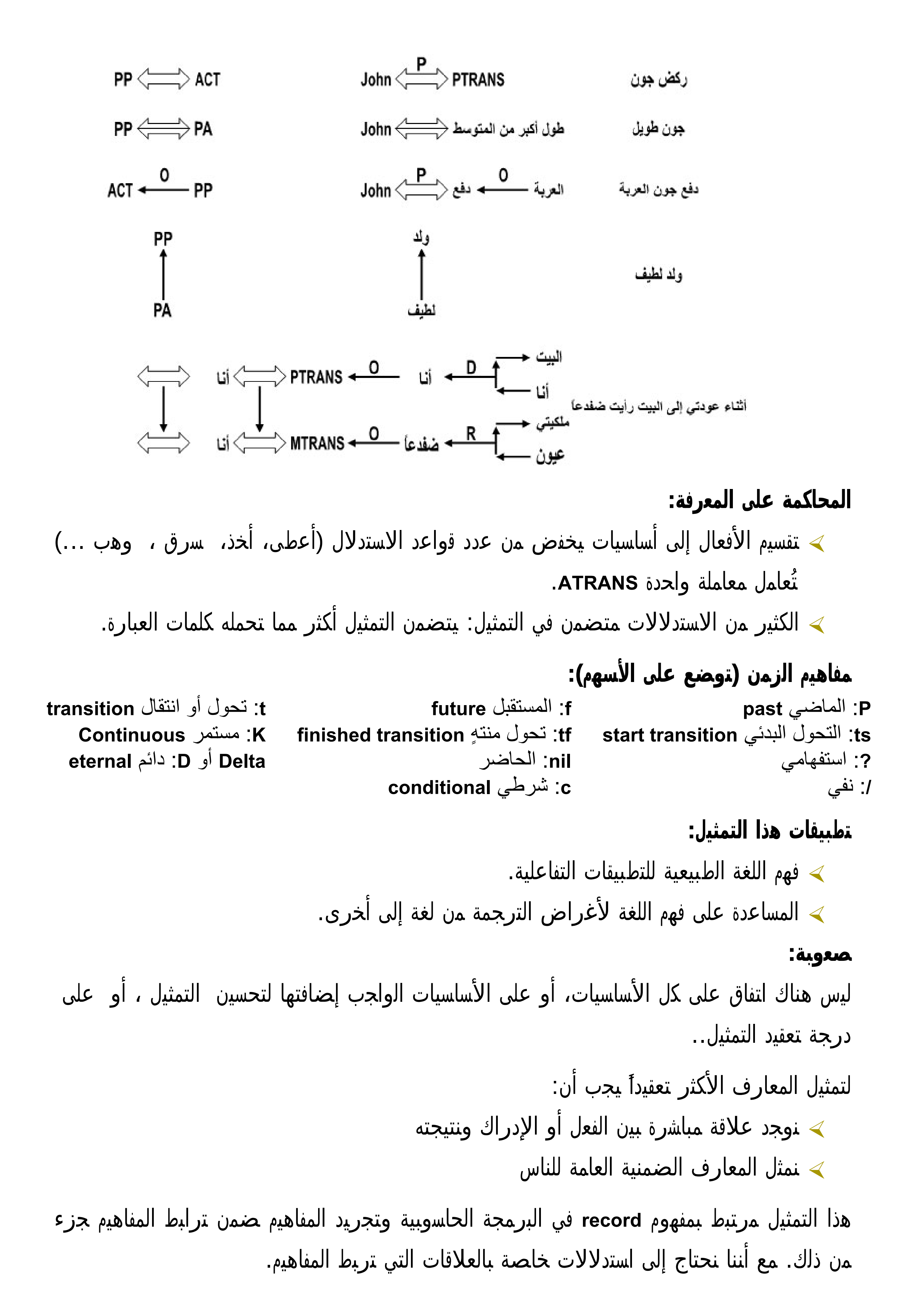
ثمة عدد من الأساسيات، يشير كل منها إلى مجموعة من الأفعال، وقد اعتمدت مجموعة الأساسيات التالية:

- **ATRANS**: **Abstract Transfer** أي نقل مجرد، مثل: أعطى **give**، أخذ **take**، سرق **steal**، نهب، أهدى،...
- **PTRANS**: **Physical Transfer** أي نقل موضع فيزيائي لغرض، مثل: ذهب **go**، أتى، سافر ، قفز،...
- **PROPEL**: تطبيق قوة مادية على غرض، مثل: دفع **push**، كسر، حطم، رفع،...
- **MOVE**: تحريك جزء من الجسم من قبل صاحبه، مثل: ركل **kick**، لوح، غمز، ضم،...
- **GRASP**: التقاط غرض من فاعل، مثل: رمى **throw**، التقط، أمسكن...
- **INGEST**: هضم شيء، مثل: أكل **eat**، شرب، تناول دواء،...
- **EXPEL**: لفظ شيء، مثل: بكى **cry**، صرخ، نادى، أطلق،...
- **MTRANS**: نقل معلومات فكرية، مثل: قال **say**، روى، شرح، نبه، أدب، احتج، أثنى،...
- **MBUILD**: استنتاج معلومات من أخرى، مثل: قرر **decide**، محص، درس،...
- **SPEAK**: إنتاج أصوات، مثل: قال، غنى **sing,** قال، صرخ، عزف،...
- **ATTEND**: التركيز بالحواس، مثل أصغى **listen**، راقب، أمعن، حضر،...

هذه الأساسيات تحدد ترابطات بين كيانات على أربعة أنواع هي:

- **ACT**: **Actions**، الأفعال، وهي تندرج تحت الأساسيات السابقة (جرى، أخذ، أعطى، بكى،...)
- **PP**: **Picture Producers**، وتدل على منتج الصورة أو الفاعل (شخص، حيوان، نبات..)
- **AA**: **Action Assistant**، مساعد على تغيير الفعل، قد يكون هو المفعول أو غيره (قوة،..)
- **PA**: **Picture Assistant**، مساعد على تغيير الصورة، وهي الصفات مثل (طول، لطف،..)

أمثلة إضافية:



المحاكمة على المعرفة:

- تقسيم الأفعال إلى أساسيات يخفض من عدد قواعد الاستدلال (أعطى، أخذ، سرق ، وهب ...) تُعامل معاملة واحدة **ATRANS**.
- الكثير من الاستدلالات متضمن في التمثيل: يتضمن التمثيل أكثر مما تحمله كلمات العبارة.

مفاهيم الزمن (توضع على الأسهم):

P: الماضي **past**
ts: التحول البدئي **start transition**
?: استفهامي
/: نفي
f: المستقبل **future**
tf: تحول منتهٍ **finished transition**
nil: الحاضر
c: شرطي **conditional**
t: تحول أو انتقال **transition**
K: مستمر **Continuous**
D: دائم أو **Delta eternal**

تطبيقات هذا التمثيل:

- فهم اللغة الطبيعية للتطبيقات التفاعلية.
- المساعدة على فهم اللغة لأغراض الترجمة من لغة إلى أخرى.

صعوبة:

ليس هناك اتفاق على كل الأساسيات، أو على الأساسيات الواجب إضافتها لتحسين التمثيل ، أو على درجة تعقيد التمثيل..

لتمثيل المعارف الأكثر تعقيداً يجب أن:

- نوجد علاقة مباشرة بين الفعل أو الإدراك ونتيجته
- نمثل المعارف الضمنية العامة للناس

هذا التمثيل مرتبط بمفهوم **record** في البرمجة الحاسوبية وتجريد المفاهيم ضمن ترابط المفاهيم جزء من ذلك. مع أننا نحتاج إلى استدلالات خاصة بالعلاقات التي تربط المفاهيم.

من الأمثلة على تمثيل معارف أعقد:

- الأطر (ملء شقوب **slots' filling**، وطرائق للملء)
- السيناريو (تسلسل مرتب زمنياً من الأطر)

4. الأطر Frames

هي نوع من شبكات الدلالة المعقدة، التي تتضمن بنية داخلية متطورة (معرفة مترفعة **meta-knowledge**)، صاحب الفكرة في إنشائها هو **Marvin Minsky** الذي قال في عام **1975**:

"حين نواجه موقفاً جديداً (أو نقوم بتغيير أساسي في رؤيتنا للمسألة) فإننا نختار من ذاكرتنا بنية تسمى الإطار، إنها إطار عمل يمكن تذكره، نسعى إلى ملاءمته ليناسب الحقيقة بتغيير تفاصيله إذا استدعى الأمر."

"When one encounters a new situation (or makes a substantial change in one's view of a problem) one selects from memory a structure called a 'frame'. This is a remembered framework to be adapted to fit reality by changing details as necessary."

للإطار اسم ومجموعة من الأزواج واصفة-قيمة. يقابل اسم الإطار عقدةً في شبكة دلالية ، أما الواصفات فتقابل أسماء الأسهم المرافقة لهذه العقدة، وتقابل القيم العقد في الطرف الثاني من الأسهم. نسمي عادة ، الأزواج واصفة - قيمة بالشقوب **slots**. ونسمي الواصفات أسماء الشقوب **slot names**، ونسمي القيم مالئات الشقوب **slot fillers**. ويبين الشكل مثالاً على إطار.

```
Frame name
Printers
      Subset_of: Office_machies
      Superset_of:(Laser_Printer,
Slots                   Ink_jet_printers)
      Energy_source: Wall_outlet
      Creator: John_Jones
      Date: 16_aug_91
   Slots names      Slots fillers
```

بالإضافة لذلك، يمكن تعريف قيم مغتفلة للشقوب **default values**، نعتمدها ما لم يأتنا ما يناقضها ، ويمكن أن نعرف حدوداً دنيا وعظمى للقيم بحسب نموذجنا الذهني، كما يمكن استدعاء إجرائيات برمجية عند تغيير قيمة الشقب أو عند الحاجة.

يمكننا أيضاً تمثيل بعض المعارف المترفعة **meta knowledge** في هذا التمثيل؛ على سبيل المثال ، تاريخُ إنشاء الإطار واسم من قام بإنشائه، وهي معارف عن الإطار، لا عن الطابعات.

أمثلة أخرى على الأطر:

هنا إطار لبطاقة أخذ مقعد في الطائرة، يتضمن الإطار واصفات (شقوب يجب ملؤها، يمكن استدعاء خوارزميات للملء، أو الإشارة إلى أطر أخرى):

- اسم شركة النقل،
- اسم الشخص،
- رقم الرحلة،
- تاريخ الرحلة،
- المقعد،
- مدينة الانطلاق،
- مدينة الوصول،
- ساعة الإقلاع،
- ورقم البوابة.

Air New Zealand Boarding Pass	
الشركة :	Air New Zealand
اسم المسافر :	Mr. White
الرحلة :	NZ 101
التاريخ :	13 Dec
المقعد :	25A
من :	Melbourne
إلى :	Christchurch
ساعة ركوب الطائرة :	1800
بوابة :	7

Quantas Boarding Pass	
الشركة:	Quantas
اسم المسافر:	Mr. Black
الرحلة:	Q 101
التاريخ:	12 Dec
المقعد:	24A
من:	Melbourne
إلى:	Sydney
ساعة ركوب الطائرة:	0600
بوابة:	4

أهمية الأطر

في النظم الخبيرة التي تعتمد القواعد، المعرفة مبعثرة في قاعدة المعرفة، وعلينا البحث عن المعرفة غير البارزة، وهذا يبطئ البحث. أما باستعمال الأطر، فإننا نستخدم معرفتنا النموذجية عن الأغراض والحالات وتكون المعرفة على شكل أغراض لها واصفات نحدد قيمها بطرق مختلفة.

يمكن أن يمثل الإطار غرضاً أو مجموعة من الأغراض المتشابهة:

- الإطار الصف **Class-Frame**: مجموعة أغراض لها الواصفات نفسها. **Class of computers**
- الإطار المنتسخ (المثال) **Instance-Frame**: يشير إلى منتسخ معين من الصف. **My_computer**
- كل إطار يعلم إلى أي صف ينتمي ولكل إطار خواص ضمنية.

CLASS: Boarding Pass	
شركة النقل:	[Str]
اسم المسافر:	[Str]
الرحلة:	[Str]
التاريخ:	[Str]
المقعد:	[Str]
من:	[Str]
إلى:	[Str]
ساعة ركوب الطائرة:	[Num]
بوابة:	[Num]

في المثال السابق: الإطار الصف هو المبين جانباً والأمثلة المعطاة هي منتسخات.

العلاقات بين الصفوف:

- يرتبط الصف الجزئي بالصف الأعم بعلاقة **is-a**.
- يمكن للإطار المنتسخ أن يعيد كتابة بعض القيم الخاصة به (مثال: النعامة من صف الطيور لكنها لا تطير).
- يبقى المنتسخ عضواً من صفه حتى وإن هتك بعض القيم النموذجية في الصف، وهذا يسمى التخصيص.

- الأطر من المستويات الأعلى تمثل مفاهيم مولدة **generic** أما من المستوى الأدنى فتمثل مفاهيم مقيدة.

العلاقات بين الأغراض:

- التعميم **Generalization**: **is-a, a-kind-of** (مثال: السيارة هي وسيلة نقل فهي من صف أعم).
- التجميع **Aggregation**: **part-of, whole-part** (مثال: المحرك جزء من السيارة).
- الربط **Association**: يصف علاقات دلالة بين مختاف الصفوف (مثال: يمتلك السيد **X** سيارة، ويمتلك السيد **X** كلب حراسة).

المحاكمة في الأطر:

- **الوراثة:** الإطار المنتسخ يرث **inherent** عن الإطار الأب (الصف)، ويمكن أن يرث من أكثر من أب (وراثة متعددة). لهذا الأمر سيئاته لأنه لا يميز بين الخواص الهامة والخواص العابرة، كما أن اتخاذ القرار صعب في حالات الوراثة المتعددة.
- **الطرائق methods والعفاريت demons!**: الطريقة إجرائية ترتبط بواصفة إطار وتنفذ حين الطلب. يمكن أن تحدد قيمة واصفة أو تنفذ حين تتغير قيمة واصفة. الـ **demon** يرتبط بواصفة عبارات القواعد **IF-THEN** وينفذ حين تتغير واصفات هذه العبارات. ثمة نوعان من الطرائق:
  - حين الحاجة **when-needed**
  - حين التغيير **when-changed**

الاستدلال في النظم المعتمدة على الأطر:

- يبحث محرك الاستدلال عن الأهداف **goals**.
- تلعب القواعد أدواراً غير كبيرة، معظم المعرفة ضمن الأطر.
- تضاف الطرائق والـ **demons** لإضافة أفعال إلى الأطر، ويمكن تحقيق الهدف بتنفيذ أي منها.
- إذا كان الاستدلال مقوداً بالمعطيات والأحداث فهو **bottom-up**.
- إذا كان الاستدلال مقوداً بالتوقعات فهو **top-down**.
- يبحث محرك الاستدلال عن القواعد (المتعلقة بالإطار **meta knowledge**) التي يكون الهدف نتيجتها ويبحث مدى صلاحيتها قاعدة تلو الأخرى **Backward Chaining**.
- الحصول على القيم من الطرائق "حين الحاجة **when needed**" إذا كان هذا مذكوراً في القواعد.
- البحث عن ترتيب طرق الحصول على قيم الواصفات (على سبيل المثال: ابحث عن القيمة من القواعد فإن فشلت نفذ طريقة "حين الحاجة"..)

مقارنة بين نظم القواعد ونظم الأطر:

في النظم الخبيرة التي تعتمد القواعد

- تمثل القواعد المعارف اللازمة لحل مشكلة.
- تلتقط كل قاعدة جزءاً من تجريبيات (كسبيات) المسألة.
- تضيف كل قاعدة معرفة جديدة إلى النظام فتجعله أذكى.
- يمكن تعديل قاعدة القواعد: بإضافة أو تغيير أو حذف قواعد.

في النظم الخبيرة التي تعتمد الأطر

- يجب أولاً تحديد البنية الهرمية للمعرفة.
- تحديد الصفوف والواصفات.
- بناء العلاقات التراتبية الهرمية بين الأطر.
- يسمح هذا البنيان بإضافة أفعال إلى الأطر عبر الطرائق **methods** والـ **demons**.

خطوات بناء نظام خبير يعتمد على الأطر:

1. تحديد المسألة ومدى مجال النظام.
2. تحديد الصفوف وواصفاتها.
3. تحديد المنتسخات.
4. تصميم الواجهة.
5. تعريف الطرائق والـ **demons** (حين الحاجة **when-needed** وحين التغيير **when changed**).
6. تعريف القواعد.
7. تقويم النظام وتوسيعه.

مثال على خطوات بناء نظام خبير يعتمد على الأطر: الشراء الذكي Buy Smart

- الخطوة الأولى: تحديد المسألة ومدى مجال النظام ابدأ بجمع معلومات عن الممتلكات (البيوت) المعروضة للبيع في المنطقة.

حدد تفاصيل مهمة بارزة:

- الموضع
- عدد غرف النوم والحمامات
- السعر
- وصف بسيط
- صورة جيدة للبيت

اسرد الأسئلة التي يمكن أن تتعرض لها:

- أعلى مبلغ يمكن رصده لشراء البيت

- نوع البيت الذي تفضله (شقة، بيت عربي،..)
- المنطقة التي تود العيش فيها
- عدد غرف النوم التي ترغب فيها
- عدد الحمامات التي ترغب فيها

➢ الخطوة الثانية: تحديد الصفوف والواصفات.

حدد الصفوف أو الأطر الصفوف:

- حدد مفهوم البيت **Property** ووصف عام لأهم مقوماته.
- ميز واصفات كل بيت: موضعه وسعره ونوعه وعدد غرف النوم، وصف، صورة،...
- تفاصيل للوصول إلى البيت (عنوان، رقم هاتف،...)

CLASS: *Property*
[Str] *Area*:
[Str] *Suburb*:
[N] *Price*:
[Str] *Type*:
[N] *Bedrooms*:
[N] *Bathrooms*:
[Str] *Construction*:
[Str] *Phone*:
[Str] *Pictfile*:
[Str] *Textfile*:
[N] *Instance Number*:

➢ الخطوة الثالثة: تعريف المنتسخات **Instances**.

أوجد المنتسخات باستخدام قاعدة معطيات أو وسط آخر للحصول على المعطيات:

INSTANCE:	Property 1
Class:	*Property*
[Str] *Area*:	دمشق
[Str] *Suburb*:	برزة
[N] *Price*:	1640000
[Str] *Type*:	شقة
[N] *Bedrooms*:	3
[N] *Bathrooms*:	1
[Str] *Construction*:	بناء حديث
[Str] *Phone*:	(03) 6226 4212
[Str] *Pictfile*:	*house01.bmp*
[Str] *Textfile*:	*house01.txt*
[N] *Instance Number*:	1

INSTANCE:	Property 2
Class:	*Property*
[Str] *Area*:	دمشق
[Str] *Suburb*:	مهاجرين
[N] *Price*:	1500000
[Str] *Type*:	شقة
[N] *Bedrooms*:	3
[N] *Bathrooms*:	1
[Str] *Construction*:	قديم
[Str] *Phone*:	(03) 6226 1416
[Str] *Pictfile*:	*house02.bmp*
[Str] *Textfile*:	*house02.txt*
[N] *Instance Number*:	2

➢ الخطوة الرابعة: تصميم الواجهات **Display**.

يجب أن تعرض واجهة عنوان التطبيق معلومات عامة في بداية كل استخدام:

- وصف عام للمسألة،
- صور بيانية موضحة،
- مع بعض معلومات حفظ حقوق الملكية.





- الخطوة الخامسة: تعريف الطرائق **methods** والـ **demons**: **When needed & When changed**.
- الخطوة السادسة: تعريف القواعد.

طرائق لجعل تطبيقنا حياً:

- باستخدام **when needed & when changed**.
- باستخدام قواعد مواءمة النماذج **pattern matching**.

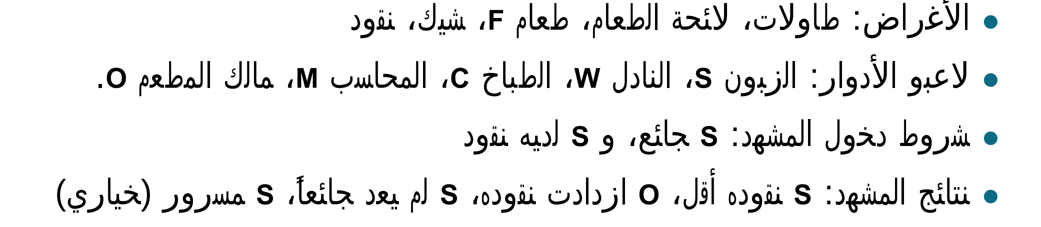
حين يعتمد نظامنا على الأطر نفضل تعريف الطرائق والـ **demons** على قواعد مواءمة النماذج.

- الخطوة السابعة: تقويم النظام وتوسيعه.
  - أحد أهم القرارات التي يجب البت بها: هل نستخدم قواعد أم طرائق؟
  - غالباً يكون القرار تبعاً لذوق المصمم.
  - المثال السابق يستخدم الطرائق لأنها فعالة وسهلة لتمثيل الإجرائيات.

5. السيناريو

المشهد **script** أو **scenario**، بنية تصف تسلسلاً من الأحداث في سياق معين **context**. لدينا شقوب **slots** ولكل شقب قيمة، وقيمة مفترضة **by-default** إذا لم نتمكن من الحصول على القيمة . ولأن المشهد يخص سياقاً معيناً، فلدينا معلومات خاصة أكثر دقة حول البنية.

- في المثال التالي (عن **Schank, 1977**) مشهد أو سيناريو للذهاب إلى مطعم:
  - عنوان المشهد **Script**: هو المطعم
  - المكان: مقهى


- الأغراض: طاولات، لائحة الطعام، طعام **F**، شيك، نقود
- لاعبو الأدوار: الزبون **S**، النادل **W**، الطباخ **C**، المحاسب **M**، مالك المطعم **O**.
- شروط دخول المشهد: **S** جائع، و **S** لديه نقود
- نتائج المشهد: **S** نقوده أقل، **O** ازدادت نقوده، **S** لم يعد جائعاً، **S** مسرور (خياري)

أما تسلسل المشاهد فيبينها **الشكل التالي**.

Script: RESTAURANT Track: Coffee Shop Props: Tables Menu F = Food Check Money Roles: S = Customer W = Waiter C = Cook M = Cashier O = Owner	Scene 1: Entering S PTRANS S into restaurant S ATTEND eyes to tables S MBUILD where to sit S PTRANS S to table S MOVE S to sitting position
	Scene 2: Ordering (Menu on table) S PTRANS menu to S (W brings menu) (S asks for menu) S MTRANS signal to W; W PTRANS W to table; S MTRANS "need menu" to W; W PTRANS W to menu W PTRANS W to table W ATRANS menu to S S MTRANS food list to CP(S) *S MBUILD choice of F S MTRANS signal to W W PTRANS W to table S MTRANS "I want F" to W W PTRANS W to C W MTRANS (ATRANS) to C
Entry conditions: S is hungry. S has money. Results: S has less money O has more money S is not hungry S is pleased (optional)	C MTRANS "no F" to W W PTRANS W to S W MTRANS "no F" to S (go back to *) or (go to Scene 4 at no pay path) C DO (prepare F script) to Scene 3
	Scene 3: Eating C ATRANS F to W W ATRANS F to S S INGEST F (Option: Return to Scene 2 to order more; Otherwise, go to Scene 4)
	Scene 4: Existing S MTRANS to W (W ATRANS check to S W MOVE (write check) W PTRANS W to S W ATRANS check to S S ATRANS tip to W S PTRANS S to M S ATRANS money to M (No pay path) S PTRANS S to out of restaurant

المثال برمجياً

Generic RESTAURANT Frame

Specialization-of: Business-Establishment
Types:
  range: (Cafeteria, Fast-Food, Seat-Yourself, Wait-To-Be-Seated)
  default: Seat-Yourself
  if-needed: IF plastic-orange-counter THEN Fast-Food,
    IF stack-of-trays THEN Cafeteria,
    IF wait-for-waitress-sign or reservations-made THEN Wait-To-Be-Seated,
      OTHERWISE Seat-Yourself.
Location:
  range: an ADDRESS
  if-needed: (Look at the MENU)
Name:
  if-needed: (Look at the MENU)
Food-Style:
  range: (Burgers, Chinese, American, Seafood, French)
  default: American
  if-added: (Update Alternatives of Restaurant)
Times-of-Operation:
  range: a Time-of-Day
  default: open evenings except Mondays
Payment-Form:
  range: (Cash, CreditCard, Check, Washing-Dishes-Script)
Event-Sequence:
  default: Eat-at-Restaurant Script
Alternatives:
  range: all restaurants with same Foodstyle
  if-needed: (Find all Restaurants with the same Foodstyle) [Rogers 1999]

برمجة المشهد (السيناريو)

EAT-AT-RESTAURANT Script

Props: (Restaurant, Money, Food, Menu, Tables, Chairs)
Roles: (Hungry-Persons, Wait-Persons, Chef-Persons)
Point-of-View: Hungry-Persons
Time-of-Occurrence: (Times-of-Operation of Restaurant)
Place-of-Occurrence: (Location of Restaurant)
Event-Sequence:
  first: Enter-Restaurant Script
  then: if (Wait-To-Be-Seated-Sign or Reservations)
    then Get-Maitre-d's-Attention Script
  then: Please-Be-Seated Script
  then: Order-Food-Script
  then: Eat-Food-Script unless (Long-Wait) when Exit-Restaurant-Angry Script
  then: if (Food-Quality was better than Palatable)
    then Compliments-To-The-Chef Script
  then: Pay-For-It-Script
  finally: Leave-Restaurant Script

[Rogers 1999]

فائدة المشاهد

- إذا لم يكن المشهد أساسياً يمكن أن يكون لدينا مؤشراً إليه دون تفعيله.مثال:
(مرت سوزان أمام مطعمها المفضل في طريقها إلى المتحف. لقد أعجبت بلوحات بيكاسو)
- حين يكون المشهد أساسياً، نستطيع استنتاج معارف جديدة دون ذكرها صراحة لأن المشهد قد عرف بنية تسلسل المعارف والحوادث، المثال التالي يجيب على سؤال "هل أكل رامي البارحة؟" بـ "نعم":(ذهب رامي إلى المطعم، وطلب شواء، وعند الدفع اكتشف أنه لم يبق لديه كثير من المال، أسرع إلى المنزل لأن المطر بدأ بالهطول)
- هو طريقة لبناء تفسيرات وحيدة متماسكة ، بدءاً من مجموعة مشاهدات . فالمشاهد أو السيناريوهات هي سلسلة علاقة سببية للأحداث، المثال التالي يجيب على سؤال "لماذا أعطت النادلة أحمد لائحة الطعام؟" بـ "لأنه طلبها منها، أو لكي يتمكن من طلب الطعام دون أن يطلبها":(ذهب أحمد إلى الغداء، جلس إلى الطاولة ونادى النادلة. أعطته النادلة لائحة الطعام ، وطلب هامبرغر)
- تسمح المشاهد بتركيز الانتباه على أحداث غير اعتيادية، مثال:(ذهب سامي إلى مطعم، اقتيد إلى طاولة، طلب شواء، بقي ينتظر مدة طويلة، غضب وغادر) هنا انقطع السيناريو بعد طلب الطعام ولا يمكننا استنتاج شيء من الأحداث التي تتلو طلب الطعام من المشهد، لحدوث شيء غير اعتيادي وهو الانتظار الطويل، حيث لا نستطيع التنبؤ بما يمكن أن يحدث بعده.

6. تمارين

1. "ذهب سامي إلى المطار، لم يكن في عجلة من أمره لأنه يتوقع تأخر الطائرة . تناول بعض الشطائر والعصير، وكم كانت مفاجأته كبيرة حين علم بأن الطائرة كانت على وقتها."
   1. مثل الحقائق السابقة مستخدماً خمساً على الأقل من المفاهيم:

ATRANS, PTRANS, MTRANS, DIGEST, MOVE, GRASP, EXPEL, ATTEND, SPEAK

   2. هل كان سامي مسافراً؟ أم ذهب إلى المطار لهدف آخر؟. أجب على ذلك من خلال سيناريو **script** تضعه لدخول المطار تبين فيه:
      - مكان المشهد،
      - أغراضه،
      - اللاعبين،
      - شروط دخول المشهد،
      - نتائج المشهد،
      - تسلسل أحداث يقوم بها المسافر،

- و تسلسل أحداث يقوم بها من يذهب لاستقبال أو وداع شخص آخر.

2. "ذهبت الأسرة لتناول البوظة، لم يرِد أحد الأطفال تناولها فرماها. تناول الآخرون البوظة وشعروا بالبرد فأسرعوا إلى السيارة وحدث بعضهم بعضاً: ما كان يجب تناولها الآن."

1. مثل الحقائق السابقة مستخدماً أربعاً على الأقل من المفاهيم:

ATRANS, PTRANS, MTRANS, INGEST, MOVE, GRASP, EXPEL, ATTEND, SPEAK

2. هل كان الجو صيفاً أم شتاء؟ أجب على ذلك من خلال سيناريو **script** تضعه لتناول البوظة في مكان عام تبين فيه فقط:

- شروط دخول المشهد،
- اللاعبون،
- نتائج المشهد،
- وتسلسل أحداث يقوم بها اللاعبون (في الصيف وفي الشتاء).

3. نود بناء قاعدة معرفة لوجبات مطاعم متعددة، بغية تحديد الوجبة المناسبة بحسب متطلبات غذائية معينة. تتضمن المعرفة عن الوجبات:

- المقبلات وما تتضمنه من خضار أو بقول أو صلصات،
- الطبق الرئيسي هل هو حيواني أم نباتي،
- طبق الحلوى (حلويات، أو فواكه أو كريمات)
- كمية الحريرات (منخفض وسط مرتفع)،

مثِّل هذه المعارف باستخدام الأطر. وأعط مثالاً **instance** لوجبة تعرفها.

7. مراجع البحث

- محاضرات نظم قواعد المعرفة: كلية المعلوماتية **2006 – 2007**، د. أميمة الدكاك
- كتاب الذكاء الصنعية رؤية جديدة
- محاضرات ورشة العمل **ANLP** انظر الموقع **www.ictis07.org**
- **What is a Knowledge Representation**

1. فضاء الحالات

تحتاج آلة تفاعلية ذات مهارة واسعة إلى كمية هائلة من الذاكرة (يمكن أن يتعذر بلوغها) لكي تستطيع أداء مهام معقدة في بيئة معقدة. إضافة إلى ذلك، يجب على مصمم مثل هذه الآلة التفاعلية أن يتمتع برؤية ثاقبة لكي يحدد ردود الفعل المناسبة لجميع الحالات المحتملة. يقودنا ذلك إلى مراعاة الزمن بدلاً من الحجم، والتكيف عوضاً عن التصميم الصريح. يمكن أن نضيف إلى وظيفة الفعل بعضاً من الحسابات التي على مصمم الآلة التفاعلية إجراؤها. بالطبع، تتطلب هذه الحسابات زمناً، ولكنها تقلّص من احتياجات الذاكرة وتخفّف الأعباء على المصمم.

لنأخذ المثال التالي:

ليكن في عالم الفضاء-الشبكي ثلاثة مكعبات **A** و **B** و **C**، تكون جميعها في البداية على الأرض. ولنفترض أن مهمة الربوط أن يكدسها بعضها فوق بعض، بحيث يكون المكعب **A** فوق **B** والمكعب **B** فوق **C** والمكعب **C** على الأرض.

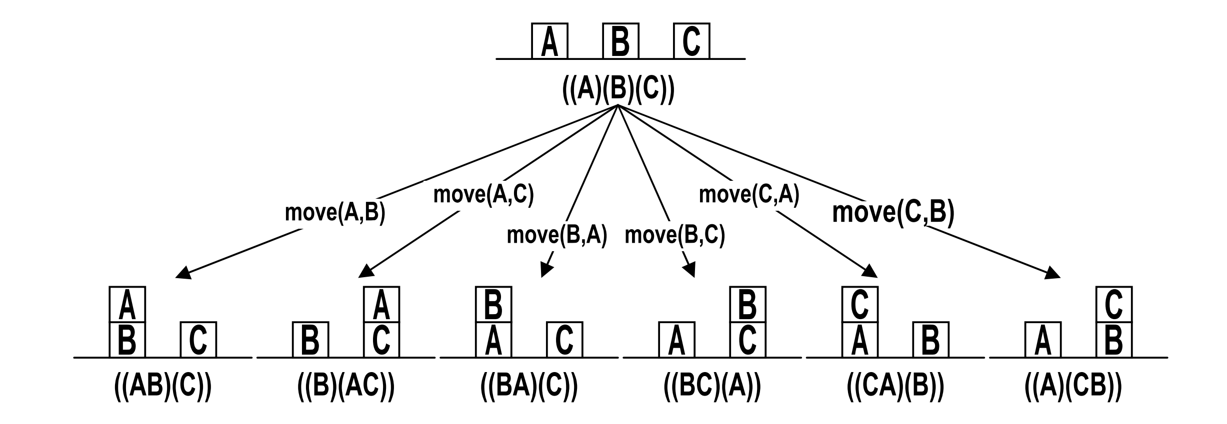
إن الأعمال الواجب أداؤها لتحقيق هذه المهمة بديهية بالنسبة لنا، في حين أنها ليست كذلك بالنسبة للربوط. لنفترض أن الربوط يستطيع نمذجة نتيجة كل فعل من أفعاله في بيئته. ويمكن أن يقوم بذلك بوساطة زوج من نماذج العالم:

- يمثل النموذج الأول حالة العالم قبل تنفيذ الفعل،
- في حين يمثل النموذج الآخر حالة العالم بعد تنفيذ الفعل.

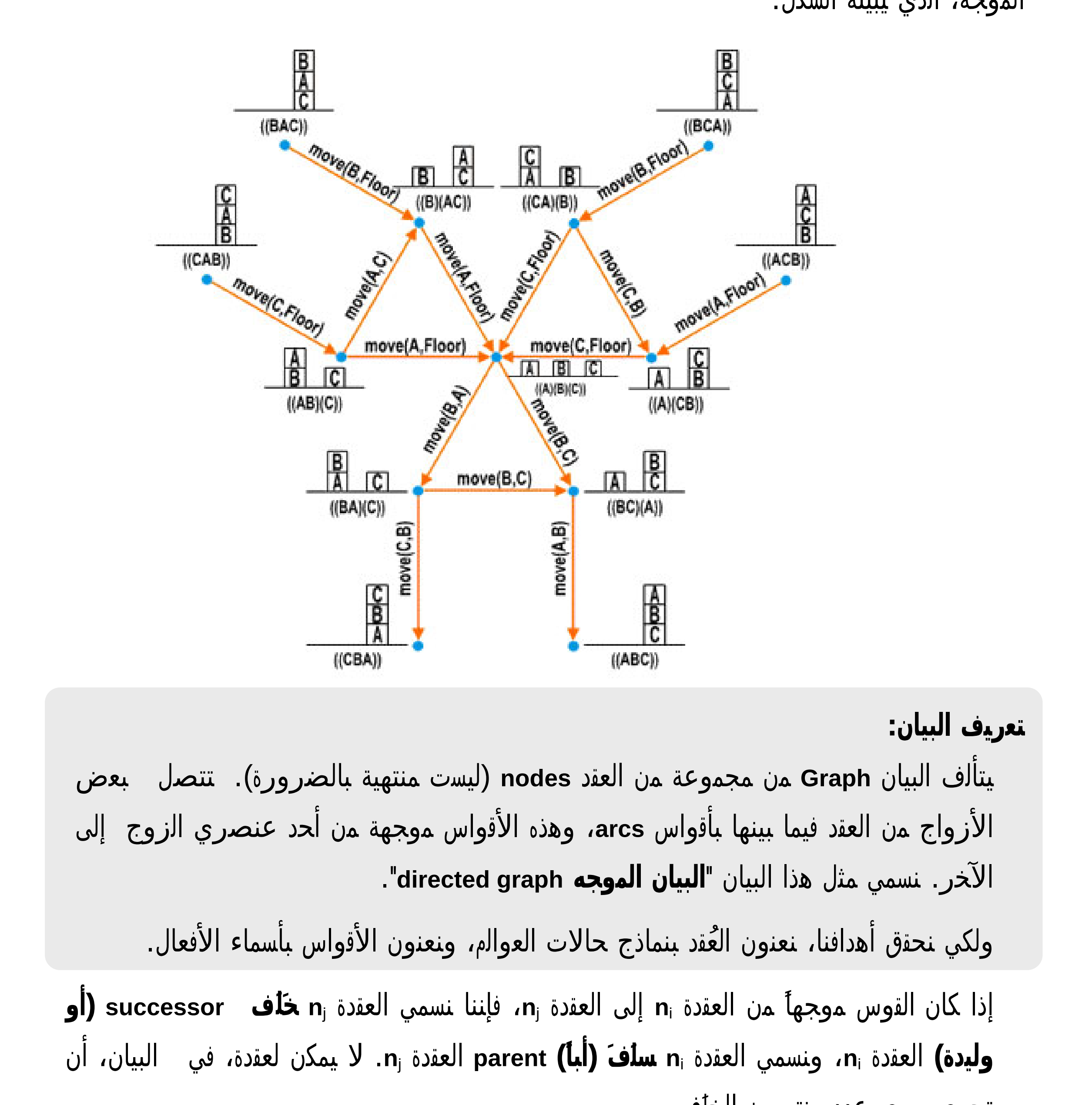
لتحقيق غاية هذا المثال، نفترض أن الربوط يستطيع تحريك أي مكعب، **x**، شريطة عدم وجود أي من المكعبات فوقه، إلى مكان آخر، **y**، إما على الأرض أو فوق مكعب لا يوجد فوقه أي مكعب آخر. تتغير البيئة لدى القيام بهذا التحريك **move(x, y)**، وهذا الفعل يسمى مؤثراً.

A B C

يبين الشكل التالي نتائج جميع الأفعال التي يمكن القيام بها في لحظة معينة.



في المحاكاة، يمكن أن ينتج عن النظر إلى الأمام خطوة واحدة فقط، توقعات مفيدة، غير أن استطلاع أكثر من خطوة، ربما على كامل الطريق حتى إتمام المهمة، يمكن أن يكتشف اختصارات ويسمح بتجنب المسارات العمياء. إن أنسب بنية لمتابعة نتائج البدائل العديدة من تسلسل الأفعال هي بنية البيان الموجه، الذي يبينه الشكل.



تعريف البيان:

يتألف البيان **Graph** من مجموعة من العقد **nodes** (ليست منتهية بالضرورة). تتصل بعض الأزواج من العقد فيما بينها بأقواس **arcs**، وهذه الأقواس موجهة من أحد عنصري الزوج إلى الآخر. نسمي مثل هذا البيان "**البيان الموجه directed graph**".

ولكي نحقق أهدافنا، نعنون العُقد بنماذج حالات العوالم، ونعنون الأقواس بأسماء الأفعال.

إذا كان القوس موجهاً من العقدة n_i إلى العقدة n_j، فإننا نسمي العقدة n_j **خَلَف successor (أو وليدة)** العقدة n_i، ونسمي العقدة n_i **سلَفَ (أباً) parent** العقدة n_j. لا يمكن لعقدة، في البيان، أن تحوي سوى عدد منته من الخلَف.

يمكن أن تكون كل عقدة في زوج من العقد خَلَفاً للأخرى؛ في هذه الحالة، نستعيض عن زوج الأقواس **بوصلة**. يسمى البيان الذي لا يحوي سوى وُصلات **بياناً غير موجه**. تحمِل الأقواس في مخططات البيان اللاحقة أسهماً، في حين لا تحمل الوصلات ذلك.

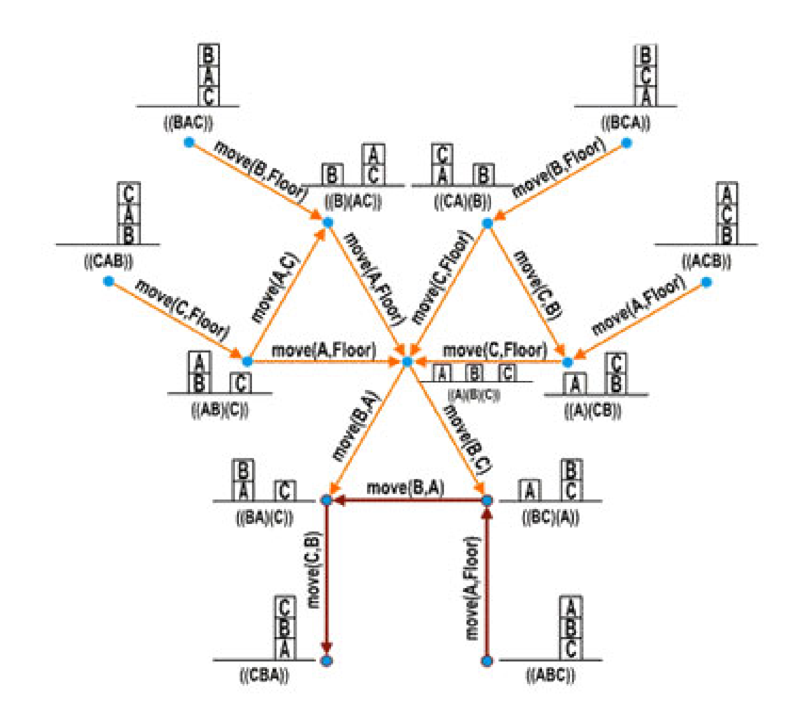
1. فضاء الحالات

إذا كان عدد حالات العوالم المختلفة صغيراً إلى حد كاف، فعندها يمكن تخزين بيان كامل يمثل جميع الأفعال والحالات الممكنة. فمثلاً، يبين البيان الموضح في الشكل السابق جميع الحالات والحركات الخاصة بالتعامل مع ثلاثة مكعبات. نسمي مثل هذا البيان لنماذج العوالم والأفعال "**بيان فضاء الحالات**". لاحظ أن كل فعل من الأفعال هو عكوس، علماً أننا عنونّا فقط أحد زوجَي الحركات المتقابلة للتنقل بين التشكيلات، وذلك لجعل الصورة أقل فوضى. وانطلاقاً من البيان، من السهل ملاحظة أنه إذا كانت الحالة الابتدائية كالتالي: **((A)(B)(C))**، وإذا كانت المهمة هي الوصول إلى الحالة الموصوفة بـ **((ABC))**، فإن على الربوط أن ينفذ تسلسل الأفعال التالي:

{move(B,C),move(A,B)}

إن من مزايا تمثيل العوالم المحتملة ببنية البيان، أنه يمكن أن نأخذ أي عقدة في البيان لتمثيل حالة هدف : من الممكن أن يكون الهدف موصفاً من مصدر خارجي كالإنسان. ولإيجاد مجموعة أفعالٍ تُو صِلُ إلى هدف محدد، لا يحتاج الربوط إلا إلى إيجاد مسار في البيان من العقدة التي تمثل حالة العالم الابتدائية (عقدة البداية) إلى العقدة التي تمثل حالة الهدف المحدد (عقدة الهدف). ومن ثم يمكن قراءة الأفعال التي تحقق هذا الهدف على اللصيقات على الأقواس في هذا المسار. فمثلاً، إذا كانت مهمة الربوط الذي يكدس المكعبات أن يكدس المكعب **C** على **B** على **A** على الأرض، وكانت الحالة الابتدائية كالآتي: المكعب **A** على **B** على **C** على الأرض، فإن تسلسل الأفعال سيكون:

{move(A,Floor),move(B,A),move(C,B)}



يجب أن نكون قادرين على تمثيل جميع حالات العوالم الممكنة بوساطة عُقَد في بيان، وعلينا معرفة النماذج الدقيقة لكيفية تنفيذ الأفعال بين زوج من حالات العوالم. يجب على نظام إدراك الربوط، في مثالنا، أن يتعرف تعرفاً صحيحاً عقدةَ البداية. كما لا يمكن أن يوجد ربوطات أو عمليات ديناميكية أخرى يمكنها تغيير العالم. فإذا تحققت جميع هذه المسلمات، وإذا كان الزمن المخصص لهذه المهمة يكفي للبحث عن حالة الهدف، فإنه من الممكن التخطيط لكامل تسلسل الأفعال ومن ثم تنفيذه (تتابعياً، كما هو).

2. البحث في فضاء الحالات

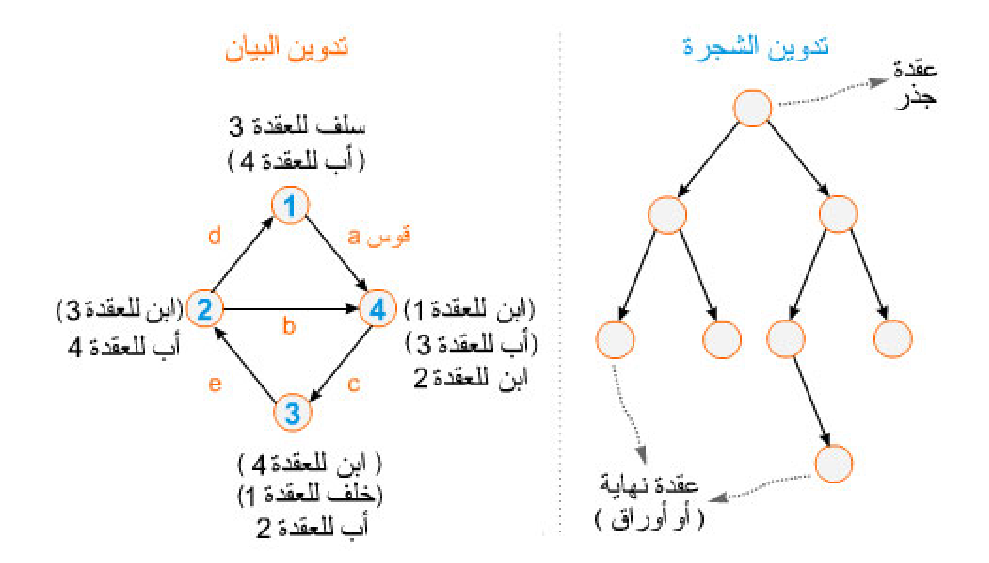
تتطلب طرائق البحث في البيانات الكاملة انتشار "مَعالِم" على عقد البيان. نبدأ بعنونة عقدة البداية بالـصفر **0**، ثم ننشر تدريجياً أعداداً أكبر على شكل أمواج على طول الأقواس إلى أن يصادف عدد ما، عقدة الهدف. ثم نرسم المسار رجوعاً من الهدف إلى البداية وفق سلسلة تنازلية من الأعداد. وتكون الأفعال على طول هذا المسار، من البداية إلى الهدف، هي الأفعال التي يجب إجراؤها لتحقيق الهدف . تحتاج هذه الطريقة إلى **O(n)** خطوة، حيث **n** هو عدد العقد في البيان.

عنونة العقد للبحث عن الهدف
(البحث الشامل)

نسمي عملية وضع علامة على خَلَفِ عقدة ما "**التوسع** expansion". يضع التوسع علامات على جميع الجوار غير المعلمة للعقدة المعلمة. إن السؤال المهم المطروح هو: ما هي العقدة المعلمة (لكن غير الموسعة) التالية الواجب توسيعها؟ بطريقة البحث عرضاً_أولاً، العقدة التالية غير الموسعة الواجب توسيعها هي تلك المعنونة بعدد لا يتجاوز جميع عناوين العقد غير الموسعة الأخرى. بمعنى آخر، نوسع جميع العقد المعنونة بالعدد **i** قبل توسيع تلك المعنونة بالعدد **j** حيث **j > i**. وثمة طرائق بحث أخرى تطبق خيارات مختلفة لتوسيع العقد؛ سوف نشرحها بالتفصيل لاحقاً.

عموماً، فإن جميع استراتيجيات البحث عن حل تعمل على توليد الحالات الواجب سبرها، ثم اختيار الحالة المرشحة للمتابعة في كل لحظة، حتى الوصول إلى الهدف (الذي قد لا يكون وحيداً).

تعتبر **الشجرة الموجهة** (ذات الجذر) حالة خاصة من البيان الموجه، لكل عقدة فيه (عدا واحدة) أب وحيد فقط. وتسمى العقدة الوحيدة التي ليس لها أب **عقدة الجذر root**. وتسمى العقد في الشجرة التي ليس لها خلف **عقد النهاية** أو **عقد الأوراق**. ونقول إن **عمق depth** عقدة الجذر يساوي **الصفر**. نعرف عمق أي عقدة أخرى في الشجرة بأنه عمق الأب مضاف إليه **1**. وتعتبر **الشجرة (ذات الجذر) غير الموجهة** (مع وصلات عوضاً عن الأقواس) بياناً غير موجه فيه مسار واحد وواحد فقط عبر الوصلات بين أي زوج من العقد.



غالباً ما يكون من المناسب وضع تكاليف موجبة على الأقواس، وذلك لتمثيل تكلفة تنفيذ الفعل المقابل. ويكون البحث عن أقصر الطرق بين عقدتين أو أقلها تكلفة.

سوف ندرس صفين واسعين من إجرائيات البحث:

- في الصف الأول، لا يوجد سبب خاص بالمسألة لتفضيل جزء من فضاء البحث على آخر، وذلك فيما يخص إيجاد مسار إلى الهدف. توصف مثل هذه الإجرائيات بالعمياء **uninformed**.
- في الصف الآخر، توجد معلومات خاصة بالمسألة للمساعدة على تركيز البحث. توصف مثل هذه الإجرائيات بالتجريبية (الكسبية) heuristic.

تأتي كلمة التجريبية (الكسبية)**heuristic** من الكلمة اليونانية **heuriskein**، وتعني " يكتشف ". وتأتي الجملة المقتبسة **Eureka!** ("وجدتها!") من المصدر نفسه.

سوف نبدأ الآن بشرح الإجرائيات العمياء ونترك الإجرائيات التجريبية إلى فقرة قادمة.

إجرائيات البحث العمياء Uninformed

نجزئ العقد المولدة إلى مجموعتين:

- **Open**: تضم عقد بيان البحث التي تنتظر المعالجة. وتحوي في البداية الحالة الأولى للمسألة.
- **Close**: تضم عقد البيان التي جرت معالجتها سابقاً.

تُطبق إجرائياتُ البحث العمياء المؤثراتِ على العقد دون استعمال أي معرفة خاصة بمجال المسألة (غير تلك المعرفة حول شرعية الأفعال). ويمكن لهذه الإجرائيات أن تعتمد إحدى طريقتين:

- البحث عرضاً_أولاً،
- البحث عمقاً_ أولاً.

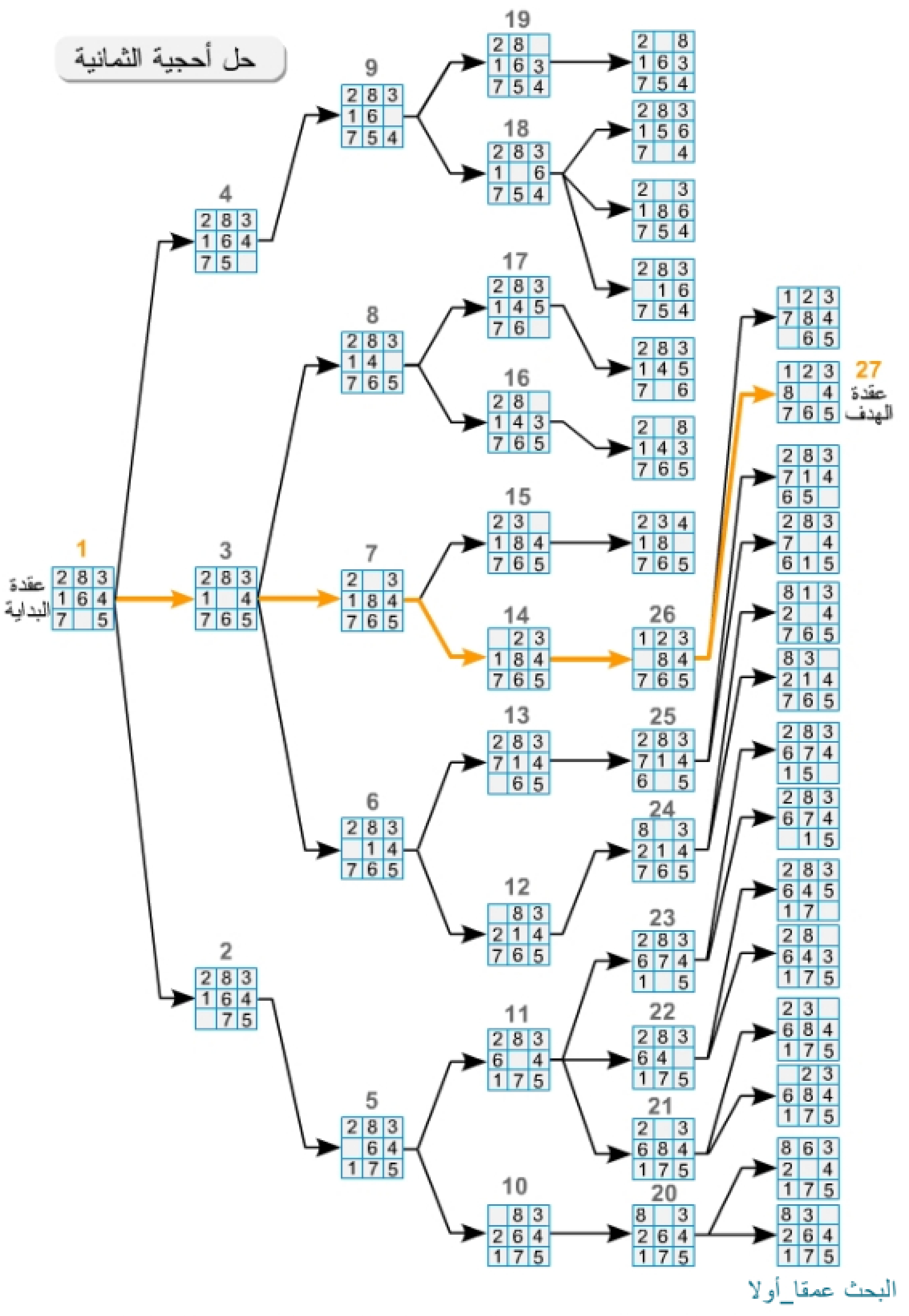
البحث عرضاً_أولاً

ربما تكون أبسط إجرائية بحث عمياء هي إجرائية البحث عرضا_أولا. تولد هذه الإجرائية بيانَ فضاء حالة شاملا بتطبيق جميع المؤثرات الممكنة على عقدة البداية، ثم بتطبيق جميع المؤثرات الممكنة على جميع خلف عقدة البداية المباشرة، ثم على خلفها، وهلم جرا. يجري البحث بانتظام نحو الأسفل انطلاقا من عقدة البداية. ولأننا نطبق جميع المؤثرات الممكنة على عقدة في كل خطوة، من المناسب ضمها جميعها في دالة تسمى دالة الخلف **Successor function**. عندما نطبق دالة الخلف على عقدة ما، فإنها تولد مجموعة العقد الكاملة التي من الممكن توليدها إذا طبقنا جميع المؤثرات القابلة للتطبيق على هذه العقدة. يسمى تطبيق دالة الخلف على العقدة: توسيع **expanding** العقدة.

للمجموعة **Open** بنية الرتل **FIFO**

يعتبر البحث ذو التكلفة الموحدة ضربا من البحث عرضا_أولا، حيث تتوسع العقد انطلاقا من عقدة البداية على طول "الحافات" ذات التكلفة المتساوية عوضا عن الحافات ذات العمق المتساوي . فإذا تطابقت تكاليف جميع الأقواس في البيان (ولنقل تساوي **1**)، فإن البحث الموحد التكلفة يصبح البحث عرضا_أولا ذاته. ويمكن اعتبار البحث الموحد التكلفة، بدوره، حالة خاصة من إجرائية البحث التجريبي (الكسبي)، الذي سوف نشرحه لاحقا.

يبين الشكل حل أحجية الثمانية للانتقال من حالة البداية إلى حالة الهدف المرتبة المبينة بالشكل بحيث ننقل في كل مرة مربعا إلى مكان فارغ مجاور.



البحث عمقا_أولا

يولد البحث عمقا_أولا خلفا واحدا لعقدة ما في كل مرة وذلك بتطبيق مؤثرات خاصة. ويُترك أثر في

كل عقدة للإشارة إلى أنه يمكن تطبيق مؤثرات أخرى إضافية عليها عند الحاجة. عند كل عقدة، يجب اختيار، المؤثر الذي سيُطبق أولا، والمؤثر الذي يليه، وهلم جرا. وما إن يولَّد الخلف، حتى يولَّد خلفه وهلم جرا. ولكي نمنع إجرائية البحث من الاستمرار إلى عقد غير محدودة العمق بالنسبة إلى عقدة البداية، نستعمل حدا للعمق **depth bound**، حيث لا يجري توليد أي خلف عمقه أكبر من حد العمق هذا. (يُفترض ألا تقع جميع عقد الهدف وراء حد العمق هذا). يسمح لنا هذا الحد بتجاهل أجزاء من بيان البحث تأكد عدم احتوائها على عقدة هدف قريبة إلى حد كاف.

للمجموعة **Open** بنية المكدس **LIFO**

من الملاحظ أن البحث "عمقا_أولا" لا يضطرنا إلا إلى حفظ جزء من شجرة البحث الذي يضم المسار المكتشف إلى الآن، إضافة إلى آثار العقد التي لم توسع كاملا بعد على طول هذا المسار. يتناسب حجم الذاكرة المطلوب في البحث عمقا_أولا، خطيا مع حد العمق. ومن مساوئ البحث عمقا_أولا، هو أن العثور على هدف ما، لا يضمن كونَ المسار إلى هذا الهدف ذا طول أصغري. تكمن مشكلة أخرى في ضرورة سبر جزء كبير من فضاء البحث حتى نجد هدفا سطحيا (ضَحلا) إذا كان هذا الهدف وحيدا وخلفا لعقدة سطحية يأتي توسيعها متأخرا في الإجرائية.

3. بيان المسائل الجزئية

أحيانا يمكن تقسيم المسألة، التي نود البحث عن حل لها، إلى مجموعة من المسائل الجزئية، بحيث يكون البحث عن حل لأحد المسائل الجزئية أسهل من المسألة الأساسية. ثم تقسيم المسائل الجزئية إلى مسائل أبسط إلى أن نصل إلى مسائل أولية بسيطة محلولة. مثل مسألة أبراج هانوي.

في هذه المسألة، لدينا ثلاثة محاور عمودية، يتوضع على المحور الأول **n** قرصا متدرجة في أقطارها، والمسألة هي نقل الأقراص إلى المحور الثالث، بحيث ننقل قرصا واحدا في كل مرة، وبحيث لا نضع أي قرص فوق قرص أصغر منه قطرا.

لعرض مثال تطبيقي لحل مسألة أبراج هانوي، قم بزيارة الموقع:

http://www.mazeworks.com/hanoi

تجزأ مسألة أبراج هانوي كما يلي:

1. نقل **n-1** قرصا من المحور الأول إلى المحور الثاني.
2. نقل القرص الكبير المتبقي من المحور الأول إلى المحور الثالث.
3. نقل **n-1** قرصا من المحور الثاني إلى المحور الثالث.

يجب تحديد كل مما يلي:

- حالات المسألة: الحالة الابتدائية والحالات العابرة.

- الهدف المطلوب الوصول إليه.
- مؤثرات التحويل المختارة.

يمكن تمثيل المسألة إما بوساطة بيان الحالات أو بوساطة بيان المسائل الجزئية. بيان الحالات درسناه سابقا، أما بيان المسائل الجزئية، فما يميزه هو أن المؤثرات ليست مؤثرات ثنائية تحول حالة إلى أخرى، وإنما مؤثرات متعددة القيمة تحول حالة إلى مجموعة من الحالات، يمثل هذا النوع من المسائل ببيان **and-or**.

يمكن أن تكون المسائل الجزئية:

- إما واجبة الترتيب، أي ترتيب حل المسائل الجزئية مفروض،
- أو مستقلة، لا يهم فيها ترتيب حل المسائل الجزئية، بل يمكن استخدام مفاهيم المعالجة التفرعية لإيجاد الحل.

4. الشجرة and-or

عند تجزئة مسألة إلى عدة مسائل جزئية، قد يكون لدينا عدة طرق لحل المسألة الأساسية **M1 or M2 or M3**، يستوجب كل منها حل مجموعة من المسائل الجزئية:

M1: OP11 and OP12 and OP13
M2: OP21 and OP22 and OP23
M3: OP31 and OP32 and OP33

في هذه الحالة تكون شجرة الحل هي شجرة **and-or**.

لا يكون حل المسألة في البيان **and-or** بيانا كاملا وإنما جزئيا بحيث يكون:

- جذر البيان الجزئي هو جذر البيان الكامل ذاته.
- لا ينطلق من عقدة **or** سوى قوس وحيد (ليصل إلى العقدة **and**)
- تنطلق من العقدة **and** جميع الأقواس الناتجة عن هذه العقدة ذاتها في البيان **and-or** الكامل ذاته. (كل أوراق البيان الجزئي هي أوراق نجاح).

إن خوارزميات البحث في البيانات **and-or** أكثر تعقيدا من البحث في بيانات الحالات لأن الحلول التي نبحث عنها ليست طرقا وإنما بيانات جزئية تتمتع بالخواص التالية:

- جذرها عقدة **or** تمثل المسألة الأساسية.
- لكل عقدة **or** ابن وحيد يمثل عقدة **and** (إحدى تجزئات المسألة الأصلية).
- لعقدة **and** كل الأبناء الخاصة بها.

مسار **or** لا يُمَثَّل بسهولة باستعمال المؤثرات **(fathers)**. الخيار بين الحلول الجزئية يعتمد غالبا على كلفة هذه الحلول، وقد يعتمد على كسبيات (تجريبيات) لإيجاد هذه الكلفة.

5. البحث التجريبي

تشبه إجرائيات البحث هنا إجرائيات البحث عرضا_أولا، غير أن البحث لا يجري بتجانس بين خلف عقدة البداية؛ وعوضا عن ذلك، يجري البحث بالمفاضلة بين العقد حيث تشير التجريبية : وهي معلومات خاصة بالمسألة، إلى العقد التي يمكن أن تكون في المسار الأفضل إلى الهدف. نسمي هذه الإجرائيات إجرائيات الأفضل_أولا أو إجرائيات البحث التجريبي. وفيما يلي الفكرة الأساسية لهذه الإجرائيات.

1. لتكن لدينا دالة تجريبية (تقويم)، $\hat{f}$، لتساعدنا على اختيار العقدة الفضلى لتوسيع العقدة التالية . تعتمد هذه الدالة على معلومات خاصة بمجال المسألة. وهي دالة تقويم حقيقية لأوصاف الحالة.
2. وسع بعد ذلك العقدة التالية، **n**، وهي تلك التي لها أصغر قيمة لـ $\hat{f}(n)$. حل الروابط اعتباطيا . (توسيع عقدة يعني إنتاج جميع خلف هذه العقدة).
3. أنهِ البحث عندما تصبح العقدة التالية المراد توسيعها هي عقدة الهدف.

يستطيع الإنسان غالبا تحديد دوال تقويم جيدة من أجل البحث بطريقة الأفضل_أولا. على سبيل المثال، في أحجية الثمانية، يمكننا محاولة استعمال عدد المربعات التي في غير مكانها مقياسا لجودة وصف الحالة:

$\hat{f}(n)$ = عدد المربعات التي في غير مكانها (مقارنة بالهدف)

يمكن أن نعتبر:

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$$

حيث:

$\hat{g}(n)$ هي تقدير "لعمق" **n** في البيان (أي، طول أقصر مسار من البداية إلى **n**)،

$\hat{h}(n)$ هي تقويم تجريبي للعقدة **n**.

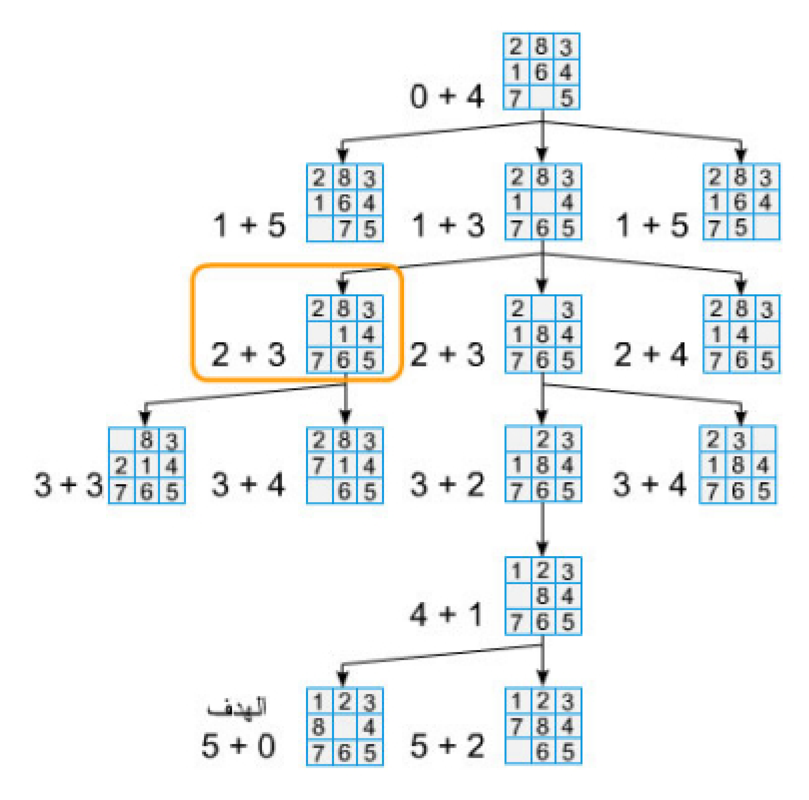
فإذا جعلنا، كما في السابق

$\hat{h}(n)$ = عدد المربعات التي هي في غير مكانها (مقارنة بالهدف)

وأخذنا

$\hat{g}(n)$ = عمق العقدة **n** في بيان البحث

حصلنا على البيان التالي:



خوارزميات البحث في بيان graphsearch

1. أنشئ شجرة بحث، Tr، مكونة فقط من عقدة البداية، n_0. ضع n_0 في قائمة مرتبة تسمى OPEN.
2. أنشئ قائمة تسمى CLOSED تكون فارغة في البداية.
3. إذا كانت OPEN فارغة، اخرُج بالنتيجة: إخفاق.
4. اختر العقدة الأولى في OPEN، واحذفها من OPEN، وضعها في CLOSED. سم هذه العقدة n.
5. إذا كانت n عقدة هدف، اخرج بنجاح مع الحل الذي حصلت عليه برسم مسار راجع على طول الأقواس في Tr من n إلى n_0. (تنشأ الأقواس في الخطوة 6).
6. وسع العقدة n، بتوليد مجموعة، M، من الخلف. أرس M باعتبارها خلفا لـ n في Tr بإنشاء الأقواس من n إلى كل عنصر من M. ضع هذه العناصر من M في OPEN.
7. أعد ترتيب القائمة OPEN، إما وَفق بعض المخططات الاعتباطية أو وَفق استحقاق تجريبي.
8. عد إلى الخطوة 3.

يمكن استعمال هذه الخوارزمية لتنفيذ بحث الأفضل_ أولا، أو البحث عرضا_ أولا، أو البحث عمقا_أولا. في البحث عرضا_أولا، نضع العقد الجديدة ببساطة في نهاية OPEN (الداخل أولا، يخرج أولا، أو FIFO)، ولا يعاد ترتيب العقد. في البحث بطريقة عمقا_أولا، نضع العقد الجديدة في بداية OPEN (الداخل أخيرا، يخرج أولا، أو LIFO). في بحث الأفضل_أولا (يسمى أيضا البحث التجريبي)،

يعاد ترتيب **OPEN** وَفق استحقاق تجريبي للعقد.

الخوارزمية A*

هي حالة خاصة من خوارزميات البحث بإستراتيجية الأفضل_أولا، تعتمد على تابع تقويم **f=h+g**. تابع التقويم **f** المطبق على عقدة **n** هو تقدير للكلفة $\hat{f}$ للطريق الأمثل من العقدة الابتدائية n_0 إلى عقدة هدف مرورا بـ **n**.

نعرف لكل عقدة **n**:

- $\hat{g}(n)$: كلفة الطريق الأمثل من n_0 إلى **n** (يساوي اللانهاية إذا لم يوجد أي طريق).
- $\hat{h}(n)$: كلفة الطريق الأمثل من **n** إلى الهدف

نستخدم تقديرا لهذه التوابع **g(n)** و **h(n)** على التتالي. ترتب القيم المرشحة للتوسيع بحسب القيم المتصاعدة لـ **f** ومن ثم التنازلية لـ **g**. يمكن أن تتغير قيم **g** أثناء البحث، حين نجد عقدا أفضل، أما **h** فسكونية.

ثمة خوارزميات تجريبية أخرى نستخدمها في برمجة الألعاب يمكن الرجوع إليها في مراجع الفصل.

6. مراجع البحث

- كتاب الذكاء الصنعي
- كتاب جامعي: "خوارزميات البحث الذكية"، تأليف د. بسام الكردي ود. باسل الخطيب ، كلية المعلوماتية.

الفصل الخامس: المعرفة والتفكير المتلبسين

لا يتوفر لدينا، في غالب الأحيان، سوى معارف غير مؤكدة عن المهام الواجب تنفيذها وعن المحيط . تسمح لنا العبارات مثل $P \vee Q$ بالتعبير عن الشك: أيهما الصحيح **P** أم **Q**؟ إلا إننا لم نصِف بعد، كيف يمكننا تمثيل درجة توثقنا **how certain** من **P** أو **Q**.

يمكننا، في المنطق العادي، استنتاج **Q** من **P** ومن $P \supset Q$. والسؤال هو، هل توجد تقانات استدلال مماثلة عندما تكون المعلومات غير مؤكدة؟ لقد استعمِلت صياغات متنوعة لتمثيل معلوماتٍ غير مؤكدة وإجراء محاكمة عليها. إن أكثر الصياغات تطورا (وقد يختلف الناس في أنها أكثر ملاءمة) هي التقانة التي تعتمد على الاحتمالات.

1. مراجعة الاحتمالات

لنفترض أن لدينا مجموعة من المتغيرات العشوائية $V_1, V_2, ..., V_k$. نشير إلى الاحتمال المشترك **joint probability**، أنْ تأخذ المتغيرات $V_1, V_2, ..., V_k$ القيم $v_1, v_2, ..., v_k$ على الترتيب، بالتعبير $p(V_1=v_1, V_2=v_2, ..., V_k=v_k)$.

نسمي التعبير $p(V_1,V_2, ...,V_k)$ دالة الاحتمال المشترك **joint probability function** للمتغيرات $V_1, V_2,..., V_k$.

مثلا، في مسألة رمي قطعة نقود متوازنة (نشير إلى أحد الوجوه بالرمز **Head (H)** وللآخر بالرمز **Tail (T)**)، يمكن أن يكون لدينا $p(H)=1/2$. فإذا رمينا قطعة النقود المتوازنة خمس مرات يمكن أن يكون لدينا مثلا:

$$p(H , T, T, H, T)=1/32$$

حيث $p(H , T, T, H, T)$ هي الاحتمال المشترك لتعطي أولُ تجربةِ رمي الحالة " وجه **H**"، والثانية الحالة "قفا **T**"، والثالثة "قفا **T**"، والرابعة "وجه **H**" والخامسة "قفا **T**".

يجب أن تحقق دوال الاحتمال المشترك خواص معينة، منها:

$$0 \leq p(V_1, V_2,..., V_k) \leq 1 \qquad (a)$$

$$\sum p(V_1, V_2,..., V_k) = 1 \qquad (b)$$

حيث تجري عملية الجمع على جميع قيم المتغيرات. وهكذا، فإنه في مثال قطعة النقود، فإن $p(H)=1/2$ متسقة مع الخصيصة **(a)**. في حين أن هذه القيمة، وبمراعاة الخصيصة **(b)**، تقتضي أن يكون لدينا $p(T)=1/2$.

تعتمد كيفية إسناد الاحتمالات إلى قيم المتغيرات العشوائية على التقدير الشخصي للخبير في مجال التطبيق (أو على معالجة إدراكية لمعطيات المحسات)، وكذلك تعتمد قيم احتمالات المتغيرات العشوائية

على تقدير الخبير أو على المعالجة الإدراكية. تقابل المتغيرات، في التطبيقات التي سنعالجها في هذا الفصل، فرضياتٍ عن المجال. يمكن أن تكون هذه الفرضيات **True** أو **False**. و من ثم تأخذ المتغيرات المقابلة القيم **True** أو **False**. يمكن أن نكون غير متوثقين من صحة واحدة أو أكثر من هذه الفرضيات؛ ونمثل عدم اليقين (الشك) هذا باحتمال قيم المتغيرات الموافقة.

سيكون من المفيد أن نؤطر عرض المفاهيم الهامة بمثال محدد. سنستخدم المثال التالي:

نأخذ ذرات الفرضيات **BAT_OK** و **MOVES** و **LIFTABLE** التي تقابل، على الترتيب، البطارية مشحونة تماما، والذراع يتحرك (حين يمسك الكتلة) والكتلة قابلة للحمل. نضيف إلى هذه الذرات الذرة **GAUGE**، التي يقصد منها أن تعني أن المقياس الذي يعطي حالة البطارية يشير إلى أن البطارية مشحونة شحنا كاملا. ولجعل مخططاتنا وصيغنا، أصغر حجما بعض الشيء، فإننا سنسمي هذه الذرات بالأحرف الأولى من أسمائها أي **B** و **M** و **L** و **G**. لنفترض الآن أننا لسنا واثقين من قيم هذه الذرات، أهي **True** أم **False**. وقبل أن نقوم بأي عملية قراءة للمجسات، لدينا معرفة سابقة باحتمالات التراكيب المختلفة لهذه القيم. فمثلا نستبعد أن تأخذ **M** قيمة **False** حين تكون بقية الفرضيات **True**.

نظرا لوجود أربعة متغيرات ذات قيم إثنانية فيوجد **16** احتمالا مشتركا لهذه المتغيرات، لكل منها الشكل:

p(B=b, M=m, L=l, G=g)

حيث تأخذ كل من **b** و **m** و **l** و **g** القيم **True** أو **False**.

على سبيل المثال، سنسرد بعضا من هذه الاحتمالات المشتركة في الجدول التالي:

(B, M, L, G)	Joint probability (الاحتمال المشترك)
(True,True,True, True)	0.5685
(True,True,True, False)	0.0299
(True,True,False,True)	0.0135
(True,True,False,False)	0.0007
(..., ..., ..., ...)	...

حين نعرف قيم جميع الاحتمالات (بالطبع قد لا يكون المصمم قادرا على تعيين الاحتمالات بمثل الدقة المعطاة بالجدول) المشتركة لمجموعة من المتغيرات العشوائية، يمكننا أن نحسب الاحتمال الهامشي **marginal probability** لأحد هذه المتغيرات العشوائية.

يعرف الاحتمال الهامشي **p(B=b)** مثلا بأنه مجموع كل قيم الاحتمالات المشتركة الثمانية من أصل القيم الست عشرة التي يكون فيها **B=b**.

$$p(B = b) = \sum_{B=b} p(B,M,L,G)$$

وباستخدام هذه الصيغة، نجد أن الاحتمال الهامشي **p(B=True)=0.95**، وهو مجموع القيم الثماني

للاحتمالات المشتركة التي يكون فيها لـ **B** القيمة **True**.

يمكن أيضا حساب الاحتمالات المشتركة من مرتبة أدنى، بحساب جمع مناسب على الاحتمالات المشتركة الكاملة. على سبيل المثال، الاحتمالُ المشترك **p(B=b, M=m)** يساوي مجموع قيم جميع الاحتمالات المشتركة الأربعة التي يكون فيها **B=b** و **M=m**:

$$p(B = b, M = m) = \sum_{B=b,M=m} p(B, M, L, G)$$

وينتج عن ذلك، أنه عندما نعرف قيم الاحتمالات المشتركة بمرتبة أقل، يمكن أن نستخدمها لحساب الاحتمالات الهامشية والاحتمالات المشتركة من مرتبة أقل. وهكذا، على سبيل المثال:

$$p(B = b, M = m) = \sum_{B=b,M=m} p(B, M, L) \quad \text{و} \quad p(B = b) = \sum_{B=b} p(B, M)$$

سنستخدم عند التعامل مع متغيرات الفرضيات (التي قيمها **True** أو **False**) تدوينا مختزلا في غالب الأحيان. على سبيل المثال، عوضا عن كتابة **p(B=True, M=False)** سنكتب $p(B, \neg M)$.

إلا أنه عندما يكون لدينا مجموعة كبيرة من المتغيرات العشوائية، فإن مهمة تحديد كل الاحتمالات المشتركة، تصبح غير قابلة للتعقب (للتحقيق) عمليا. ولحسن الحظ، تحقق الاحتمالات المشتركة، في أكثر التطبيقات، شروطا خاصة معينة تمكّن من تحديدها وإجراء حسابات عليها.

نريد أن نكون قادرين على استخدام المعلومات المتوفرة عن قيم بعض المتغيرات للحصول على احتمالات قيم بعض المتغيرات الأخرى. مثلا، إذا كان ربوط حمل الكتل يحِس أن ذراعه لا يتحرك، فربما يريد (مع معرفة هذه الحقيقة) حساب احتمال كون البطارية مشحونة. يسمى هذا النوع من الحسابات بالاستدلالات الاحتمالية **Probabilistic Inference** بالمماثلة مع طرائق الاستدلال المنطقية.

ندون دالة الاحتمال الشرطي لـ V_i بفرْض أن V_j معطاة بـ $p(V_i|V_j)$ (مهما كانت قيم المتغيرات V_i و V_j)، ويعطى بـ:

$$p(V_i|V_j) = p(V_i,V_j)\ p(V_j)$$

حيث $p(V_i,V_j)$ هو الاحتمال المشترك لـ V_i و V_j، و $p(V_j)$ هو الاحتمال الهامشي لـ V_j. ونرى من هذا التعبير أنه يمكننا أيضا كتابة الاحتمال المشترك بدلالة الاحتمال الشرطي:

$$p(V_i,V_j) = p(V_i|V_j)\ p(V_j)$$

بالعودة إلى مثال حمل الكتل، يمكن أن نحسب احتمال كون البطارية مشحونة علما أن الذراع لا يتحرك:

$$p(B = True \mid M = False) = p(B = True, M = False) / p(M = False)$$

يمكن حساب البسط **p(B = True, M = False)** والمقام **p(M = False)** لهذا التعبير بجمع الاحتمالات

المشتركة المناسبة.

يمكن فهم الاحتمالات الشرطية باستعمال تفسير الاحتمال على أساس التواتر **frequency**. فتكون **p(M=False)** على سبيل المثال، هي نسبة عدد المرات التي لا تتحرك فيها الذراع إلى مجموع عدد المحاولات (في تجارب تخيلية تجرى عددا لا نهائيا من المرات). وعلى هذا فإن احتمال أن تكون البطارية مشحونة مع كون الذراع لا تتحرك يعطى بعدد المرات التي لم تتحرك، فيها الذراع مع كون البطارية مشحونة، مقسوما على عدد المرات التي لم تتحرك فيها الذراع. وبذلك، يكون الاحتمال الشرطي نسخة مقيسة عن الاحتمال المشترك. نلاحظ أن:

$$p(B) = p(B,M) + p(B,\neg M)$$

يمكننا أيضا الحصول على الاحتمالات الشرطية المشتركة لعدة متغيرات، مشروطة بعدة متغيرات أخرى. على سبيل المثال (باستعمال التدوين المختزل):

$$p(\neg G,B \neg M,L) = p(\neg G,B,\neg M,L) / p(\neg M,L)$$

يمكن أيضا التعبير عن الاحتمال المشترك بدلالة سلسلة **chain** من الاحتمالات الشرطية. مثلا:

$$p(B,L,G,M) = p(B|L,G,M)\ p(L|G,M)\ p(G|M)\ p(M)$$

ولأن الطريقة التي نرتب فيها المتغيرات في دالة الاحتمال المشترك غير مهمة (بشرط الاحتفاظ بهذا الترتيب، ومعرفة كل العناصر) يمكننا أن نكتب ما يلي:

$$p(V_i,V_j) = p(V_i|V_j)\ p(V_j) = p(V_j|V_i)\ p(V_i) = p(V_j,V_i)$$

or

$$p(V_i|V_j) = p(V_j\ |V_i)\ p(V_i)\ /\ p(V_j)$$

هذه المعادلة الأخيرة هامة جدا وتدعى بقاعدة بايز **Bayes' Rule**.

كان بايز **Reverend Thomas Bayes [Bayes 1763]** أول من صاغ هذه القاعدة.

2. قاعدة بايز والمحاكمة باستخدامها

قاعدة بايز

$$P(A/B) = P(B/A) \cdot P(A) / P(B)$$

حيث:

- **P(A)** احتمال وقوع **A**.
- **P(B)** احتمال وقوع **B**.
- **P(A/B)** و **P(B/A)** احتمالات شرطية

وذلك بافتراض **A** و **B** غير متعارضتين.

في حال كان حدوث **A** مشروطا بحدثين متعارضين **B** و **¬B** فقط. يكون لدينا:

$P(A)= P(A/B) \cdot P(B)+ P(A/\neg B) \cdot P(\neg B)$

بالمشابهة إذا كان حدوث **B** مشروطا بحدثين متعارضين **A** و **A¬** فقط. يكون لدينا:

$P(B)= P(B/A) \cdot P(A)+ P(B/\neg A) \cdot P(\neg A)$

بتعويض العلاقة السابقة في حساب **P(A/B)** يمكن أن نكتب:

$P(A/B) = P(B/A) \cdot P(A)/ P(B) = P(B/A) \cdot P(A) / [P(B/A) \cdot P(A)+ P(B/\neg A) \cdot P(\neg A)]$

أمثلة على تطبيق قاعدة بايز

مثال (1):

احتمال النجاح في مادة الذكاء الصنعي هو

P(A)=0.7

احتمال الحصول على علامة جيدة بالعملي هو

P(B)=0.8

احتمال أن تكون علامة العملي جيدة علما أن الطالب نجح في الامتحان

P(B/A)=0.9

ما هو احتمال النجاح في الامتحان إذا كانت علامة العملي جيدة؟

P(A/B)?

من القاعدة $P(A/B) = P(B/A) \cdot P(A) / P(B)$ نجد أن: $P(A/B)=0.9 \cdot 0.7/0.8$

مثال (2):

احتمال ارتفاع الحرارة

P(A)

احتمال الإصابة بالكريب هو

P(B)=0.3

احتمال ارتفاع الحرارة بوجود الكريب

P(A/B)=0.8

احتمال ارتفاع الحرارة بغياب الكريب

P(A/¬B)=0.4

وجود الكريب أو غيابه حدثان متعارضان.

احسب احتمال ارتفاع الحرارة.

P(A)?

نطبق قاعدة حدوث **A** مشروطا بحدثين متعارضين **B** و **B¬** فقط:

$P(A)= P(A/B) \cdot P(B)+ P(A/\neg B) \cdot P(\neg B)$

المحاكمة

لنفترض إنه إذا كانت الفرضية **H** محققة وقع الحدث **E** باحتمال **P**. فنعرف:

الاحتمال القبلي **Prior Prob.:** الاحتمال غير المشروط للحدث بدون أي دليل على حدوثه أو عدمه **P(Event)**.

الاحتمال البعدي **Posterior Prob.**: الاحتمال الشرطي لوقوع الحدث بوجود بعض الأدلة **P(Event/ Evidence)**.

في حالة التشخيص نود حساب احتمال صحة فرضية التشخيص H_i اعتماداً على الأدلة (الأعراض) E_j. حيث لدينا الاحتمالات القبلية $P(H_i)$ والاحتمالات الشرطية $P(E_j/H_i)$ ونود حساب $P(H_i/E_j)$. فيكون:

$$p(H \mid E) = \frac{p(E \mid H).p(H)}{p(E \mid H).p(H) + p(E \mid \neg H).p(\neg H)}$$

في حالة النظم الخبيرة يحدد الخبير الاحتمالات القبلية $p(H), p(\neg H)$ والاحتمالات الشرطية $p(E \mid H), p(E \mid \neg H)$ ويجري حساب $p(H \mid E)$ من العلاقة السابقة. (فرضية واحدة ودليل واحد). إذا كان لدينا m فرضية $H_i, i=1,...m$ و n دليل $E_j, j=1,...n$، يجب أن تكون الأدلة متعارضة مثنى، وأن تكون الفرضيات شاملة أي:

$$p(H_i \mid E_1,...,E_n) = \frac{p(E_1,...,E_n \mid H_i).p(H_i)}{p(E_1,...,E_n)}$$

$$p(H_i \mid E_1,...,E_n) = \frac{p(E_1 \mid H_i).p(E_2 \mid H_i)....p(E_n \mid H_i).p(H_i)}{\sum_j p(E_1 \mid H_j).p(E_2 \mid H_j)....p(E_n \mid H_j).p(H_j)}$$

إذا كانت الأدلة مستقلة مثنى.

مثال على المحاكمة

نود دراسة احتمال نجاح طالب في مادة ما. لدينا الفرضيات التالية:

H_1: النجاح في العملي وفي الامتحان.

H_2: النجاح في العملي والرسوب في الامتحان.

H_3: الرسوب في المادة.

أما الأدلة أو الملاحظات التي يمكن مراقبتها فهي:

E_1: حضور جميع جلسات العملي.

E_2: قضاء 3 ساعات يوميا في الكافتيريا.

E_3: دراسة المراجع المتعلقة بالمادة.

	Hypothesis		
	i = 1	i = 2	i = 3
$P(H_i)$	.70	.10	.20
$P(E_1 \mid H_i)$	.80	.50	.10
$P(E_2 \mid H_i)$	.30	.50	.70
$P(E_3 \mid H_i)$	.70	.30	.05

أحد الخبراء بالمادة، أعطى الاحتمالات القبلية والشرطية المبينة في الجدول، والمطلوب حساب:

- $P(H_1/E_1)$ أي احتمال النجاح في العملي والامتحان بحضور جميع جلسات العملي.

$P(H_1/E_1)=P(E_1/H_1).P(H_1)/ PP$

حيث:

$PP = P(E_1/H_1). P(H_1) + P(E_1/H_2). P(H_2) + P(E_1/H_3). P(H_3)$

بالحساب نجد:

$PP=.7*.8+.1*.5+.2*.1=.63$

- $P(H_1/E_1)=.56/.63=.889$ $P(H_3/E_2)$ أي احتمال الرسوب في المادة مع قضاء 3 ساعات في الكافتيريا يوميا.

$P(H_3/E_2)=P(E_2/H_3).P(H_3)/ PP_1$

$PP_1 = P(E_2/H_1). P(H_1) + P(E_2/H_2). P(H_2) + P(E_2/H_3). P(H_3)$

بالحساب نجد:

$PP_1=.7*.3+.1*.5+.2*.7=.4$

- $P(H_3/E_2)=.14/.40=.35$ $P(H_1/E_1.E_3)$ أي احتمال النجاح بالعملي والامتحان مع حضور جلسات العملي ودراسة المراجع.

$P(H_1/E_1.E_2)=P(E_1/H_1).P(E_3/H_1).P(H_1)/ PP_2$

$PP_2 = P(E_1/H_1). P(E_3/H_1).P(H_1) + P(E_1/H_2).P(E_3/H_2). P(H_2) + P(E_1/H_3). P(E_3/H_3). P(H_3)$

بالحساب نجد:

$PP_2=.392+.015+.001$

$P(H_1/E_1.E_3)=.392/PP2=.961$

تمرين: احسب احتمال النجاح بالعملي والامتحان في حال قضاء 3 ساعات يوميا في الكافتيريا.

حسنات ومساوئ المحاكمة باستخدام شبكات بايز

حسناتها:

- لها أساس نظري.
- لها مدلول واضح حين اتخاذ القرار.

مساوئها:

- تحتاج إلى معطيات كثيرة وحسابات احتمالات.
- قد لا تكون الأدلة الشخصية واحتمالاتها مقنعة.
- الأدلة ليست دائما مستقلة.
- الحسابات طويلة.
- العلاقة بين الأدلة والفرضيات تختصر إلى أرقام.

3. الشك

يعرف الشك بأنه نقص في المعرفة الدقيقة التي تمكننا من الوصول إلى نتائج موثوقة. هذا النقص يمنع النظم الخبيرة على سبيل المثال من اتخاذ القرارات الصائبة.

عوامل اليقين هي تقنية استخدمت في عدد من النظم الخبيرة على رأسها **MYCIN** و **PROSPCTOR**. الفكرة هي أنه يمكن الوصول إلى نتائج أفضل باستخدام عدد أكبر من الاختبارات. هذه الاختبارات قد

تكون مكلفة بالزمن والمال.

يمكن أن يأتي الشك من معطيات محسات غير موثوقة، والإستراتيجية هي أنه من الأفضل اعتماد معطيات تقريبية عوضا عن انتظار المعطيات الدقيقة.

Ray Simpson (1944)		Milton Hakel (1968)	
Term	Mean value	Term	Mean value
Always	99	Always	100
Very often	88	Very often	87
Usually	85	Usually	79
Often	78	Often	74
Generally	78	Generally	74
Frequently	73	Frequently	72
Rather often	65	Rather often	72
About as often as not	50	About as often as not	50
Now and then	20	Now and then	34
Sometimes	20	Sometimes	29
Occasionally	20	Occasionally	28
Once in a while	15	Once in a while	22
Not often	13	Not often	16
Usually not	10	Usually not	16
Seldom	10	Seldom	9
Hardly ever	7	Hardly ever	8
Very seldom	6	Very seldom	7
Rarely	5	Rarely	5
Almost never	3	Almost never	2
Never	0	Never	0

مصادر الشك

- قد لا يكون الخبير قادرا على اكتشاف الترابط الوثيق بين الجزء الـشرطي **(IF part)** من قواعده ونتائج هذه القواعد **(THEN part)**.
- أحيانا يستخدم الخبير اللغة الطبيعية التي لا تخلو من التباس في التعبير عن معرفته (مثل أحيانا، غالبا، نادرا...). ولا يمكن تكمية هذه الواصفات بأرقام ثابتة تدل على نسبة اليقين بها. وإذا كان لدينا عبارة (أغلق الكتاب) ليس شرطا أن يكون واضحا أي كتاب نقصد.
- قد تكون بعض المعطيات غير معروفة أو ناقصة. فعبارة (أدر المفتاح) لا تعني بالضرورة أننا نعرف أي اتجاه نقصد، يجب إضافة مع عقارب الساعة أو عكسها مثلا أو إلى اليمين أو اليسار.
- قد نضطر إلى ضم خبرات أكثر من خبير في مجال ما.
- قد يكون لدينا أخطاء تكرارية (من حساس أو بنتيجة دقة حسابات)، خطأ بقراءة حساس أو في دقة القياس.
- قد يكون لدينا خطأ في المحاكمة (الاستنتاج والاستقراء ..)

مع أن الاستنتاج سليم عادة إلا أن قواعدنا يمكن ألا تكون سليمة في صياغتها الأصلية فنستنتج نتائج خاطئة، وقد تكون القاعدة صحيحة في غالب الأحيان ولكن ليس في كل الأحيان . فنعمم ولا نكون متأكدين من صحة التعميم.

4. عوامل اليقين Certainty Factors

استخدمت عوامل اليقين في **Mycin** وهي أرقام تدل على اعتقاد الخبير بفرضية معينة:

- الرقم **+1** يدل على يقين بالصحة.
- الرقم **-1** يدل على يقين بالخطأ.
- الرقم **0** تدل على غياب اليقين بصحة أو خطأ الفرضية.

الرقم	دلالته اللغوية
-1	أكيد لا
-0.8	تقريباً أكيد لا
-0.6	من المحتمل لا
-0.4	ربما لا
-0.2 to 0.2	غير معروف
+0.4	ربما نعم
+0.6	من المحتمل نعم
+0.8	تقريباً أكيد نعم
+1	أكيد نعم

- الرقم الموجب يدل على وجود أدلة تدعم الفرضية.
- الرقم السالب يدل على وجود أدلة تدحض الفرضية.

ويبين الجدول ترجمة مفردات اللغة الطبيعية بحسب هذه الأرقام.

يمكن أن تكون الحقائق مشكوك بها أو القواعد أو كليهما. ويجب حساب عوامل اليقين للنتيجة (نتيجة القواعد) بناء عليها.

حساب معامل اليقين لمقدمة قاعدة

إذا كانت المقدمة هي مجموعة حقائق يفصل بينها **AND** فمعامل اليقين هو أصغر معامل يقين لهذه الحقائق.

CF(A and B) = min (CF(A), CF(B))

إذا كانت المقدمة هي مجموعة حقائق يفصل بينها **OR** فمعامل اليقين هو أكبر معامل يقين لهذه الحقائق.

CF(A or B) = max (CF(A), CF(B))

معامل اليقين لنفي حقيقة هو نفي معامل اليقين للحقيقة.

CF(¬A) = - CF(A)

مثال: لنفترض أن **CF(A) = 0.3** ،**CF(B) = 0.5** ،**CF(C) = 0.4** ،**CF(D) = -0.7** فيكون:

CF(A and B or C and not(D)) = max(min(0.3,0.5),min(0.4,0.7))=max(0.3,0.4)=0.4

لنفترض أن قاعدة تقول: إذا كانت السماء صافية فيكون التنبؤ الجوي بجو صحو، وأن معامل اليقين لهذه القاعدة **0.6**، فإذا كان معامل اليقين لكون السماء صافية هو **0.5** فإن معامل اليقين ليكون الجو صحو هو: **0.6×0.5=0.3**.

لنفترض الآن ما يلي: $P1 \Rightarrow Q1$ ،$P2 \Rightarrow Q1$، وأننا حسبنا معامل اليقين لـ Q_1 من القاعدة الأولى هو cf_1، وأن معامل اليقين لـ Q_1 من القاعدة الثانية هو cf_2، فعندها يمكن حساب معامل اليقين النهائي لـ Q_1 يعطى بالعلاقة التالية:

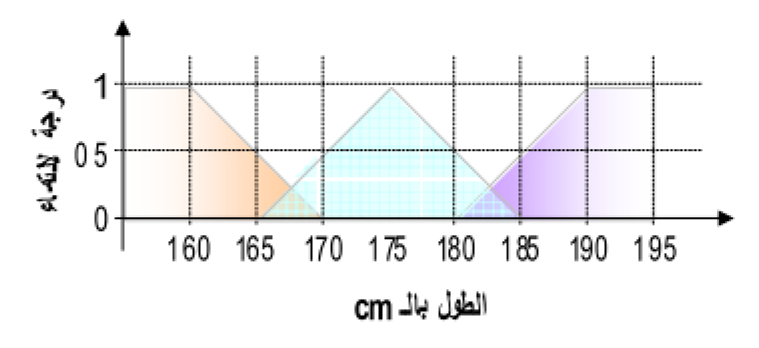
$$cf(cf_1,cf_2) = \begin{cases} cf_1 + cf_2 \times (1 - cf_1) & \text{if } cf_1 > 0 \text{ and } cf_2 > 0 \\ \dfrac{cf_1 + cf_2}{1 - \min[|cf_1|,|cf_1|]} & \text{if } cf_1 < 0 \text{ or } cf_2 < 0 \\ cf_1 + cf_2 \times (1 + cf_1) & \text{if } cf_1 < 0 \text{ and } cf_2 < 0 \end{cases}$$

5. المنطق العائم

يهتم المنطق العائم باستخدام قيم عائمة تلتقط معاني الكلمات والمحاكمات البشرية واتخاذ القرار. وهي مثلا حين يستخدم الخبير كلمات مثل عادة، على الأغلب، نادرا...الخ. ويجري استخدام قواعد شرطية إلا إنها عائمة مثل:

إذا كانت السرعة عالية فمسافة التوقف طويلة

إذا كانت السرعة منخفضة فمسافة التوقف قصيرة.



لقد ظهر المنطق العائم ونظرية المجموعات العائمة في سيتينيات القرن الماضي إلا أن استخدامه استغرق حوالي عشرين سنة . وله تطبيقات عديدة وهامة في التحكم والطب. وهذه التقنية تحسن القوة الحسابية ونمذجة المعرفة وخاصة من عدة خبراء.

المجموعات العائمة

في نظرية المجموعات التقليدية **crisp set** إذا أخذنا مجموعة **X** وأخذنا عنصرا ما **x** فإن لدينا إحدى حالتين إما $x \in X$ أو $x \notin X$. ولكن، لننظر في المثال التالي:

قال الفيلسوف اليوناني "جميع اليونانيين كذابون"، فهل هو صادق؟ في المنطق التقليدي هذه العبارة فيها تناقض أما في المنطق العائم فإن الفيلسوف صادق وكاذب بآن. فالحدود بين الصدق والكذب عائمة.

المجموعة العائمة هي مجموعة ذات حدود عائمة، نعرف لها تابع مميز

$\mu_A(x) : X \to [0,1]$

$\mu_A(x) = 1$ **if x is totally in A;**

$\mu_A(x) = 0$ **if x is not in A**

$0 < \mu_A(x) < 1$ **if x is partly in A**

مثال:

نقول عن رجل أنه قصير تماما إذا كان طوله أقل من **160 cm**، أما إذا كان طوله بين **160 cm** و **170 cm** فيتناقص انتماؤه إلى فئة الرجال القصار من **1** إلى **0**.

نقول عن رجل أنه طويل تماما إذا كان طوله أكثر من **190 cm**، أما إذا كان طوله بين **180 cm** و **190 cm** فيزداد انتماؤه إلى فئة الرجال الطوال من **0** إلى **1**.

الطول المتوسط تماما هو **175 cm**، ويتناقص الانتماء إلى فئة متوسطي الطول يسارا حتى يصل إلى **165 cm** ويمينا حتى يصل إلى **185 cm**.

وتكون القواعد عائمة أيضا، لننظر في الأمثلة التالية:

- إذا كانت مدة المشروع طويلة، وفريق العمل كبيرا والتمويل غير مناسب فالمخاطرة عالية.
- إذا كانت الخدمة ممتازة والطعام لذيذا فإن الإكرامية جيدة.

وهناك طرق لاستنتاج معارف جديدة (عائمة) بتطبيق مثل هذه القواعد.

لمزيد من المعلومات انظر كتاب **Negnevisky**.

6. مراجع البحث

- محاضرات في نظم قواعد المعرفة: كلية المعلوماتية س4 ذكاء صنعي. د. أميمة الدكاك و د. ندى غنيم
- كتاب artificial Intelligence: a guide to intelligent systems للكاتب Michael Negnevisky الطبعة الثانية للناشر Addison Wisley عام 2002.

الفصل السادس: التعلم

1. مقدمة

من الأمور التي تميز الكائنات البشرية هي إمكان التعلم. فالطفل يتعلم اللغة من الوسط الذي يعيش فيه، والعامل يتعلم ممن هو أقدم منه أو من دورات تعليمية وغيرها الشيء الكثير. فهل يمكن أن نزود الآلة بإمكان التعلم؟

استراتيجيات التعلم

التعلم في الذكاء الصنعي:

> هو تحصيل المعرفة الصريحة. أي إمكان تحصيل المعرفة وإعادة صياغتها لتصبح صريحة . ويمكن أن نعرف التعلم بأنه تحصيل نموذج للعالم حولنا.

تطور مفهوم التعلم تاريخيا وفق المسار التالي:

بدأنا بفكرة أن من يتعلم يجب أن تكون لديه معرفة. في الصناعة يمكن أن نعلم ربوطا كيف يدهن آلة ما، نمسكه باليد وندهن فيتعلم ويدهن وحده (يمكن رسم ذلك)، يتطلب هذا التعلم ذاكرة فقط.

ثم ظهر التعلم بواسطة إرشادات المعلم (المعلم بشر أو برنامج): حين يجد المعلم خطأ يوقف النظام ويصحح الخطأ، بصياغة قواعد على القواعد **meta rules**.

ثم ظهرت الشبكات العصبونية، وهي برمجيات تكرارية فيها متحولات هامة، في كل دورة نقيس درجة النجاح ونغير في هذه المتحولات لتحسين النجاح.

ثم ظهرت في السيتينيات نظم تتعلم المفاهيم (نعلم النظام أمثلة عن المفهوم وأمثلة معاكسة له، مفهوم الطاولة: نعطيه صور لطاولات وصور لأشياء أخرى ليست طاولات). الأمثلة تفيد التعميم والأمثلة المعاكسة تفيد التخصيص إلى أن نصل إلى وصف أصغري للمفاهيم. ثم ظهر التجميع المفاهيمي في صفوف تنطلق من العام إلى الخاص.

التعلم بواسطة التفسير يعتمد على قواعد الاستدلال الاستنتاجية والاستنباطية. لدينا مثال واحد، ومعرفة حول المجال المدروس، نستخدمها باستمرار لبناء تفسير يبين ملاءمة المثال للمفهوم المراد تعلمه ولاشتقاق توصيف عام للمفهوم انطلاقا من تعميم التفسير.

ثم ظهر التعلم بالقياس أو بالمماثلة، فيتعرف النظام المتشابهات بين المفهوم الذي يود تعلمه ومفهوم معروف ليحدد الميزات التي تبقى من المفهوم المعروف والميزات المضافة. أصبح بإمكان النظام معرفة ما يجب تعلمه.

لدى بناء أي نظام تعلم، يجب تحقيق الخطوات التالية:

- كشف الحاجة إلى التعلم. مثال الحاجة إلى بناء برنامج يلعب الشطرنج.

- تحصيل المعلومات واختيار المفيد منها. نعطي النظام قواعد اللعب ونتركه يلعب، حين يخسر نعود إلى الخطوات التي سببت الخسارة.
- تحويل المعلومات إلى معرفة. صياغة قواعد على القواعد لتحسين الأداء.
- مكاملة المعارف واستخدامها في النظام. النظام أصبح جاهزا للعب بنفسه.

نظرا لأهمية النظم المبنية على القواعد، والجهود البشرية الكبيرة المطلوبة لاستنباط واستخراج قواعد جيدة من الخبراء، من الطبيعي أن نتساءل عن إمكان تعلم قواعد النظم الخبيرة آليا.

يوجد نوعان أساسيان من طرق التعلم:

- التعلم الاستقرائي **Inductive**،
- والتعلم الاستدلالي (الاستنتاجي) **Deductive**.

ويُمكن استخدام كلا الأسلوبين في تعلم القواعد.

2. التعلم الاستقرائي: الشبكات العصبونية

هي إحدى الطرائق التي تمكّن الآلات من أن تتعلم، وذلك بالتعرض لمجموعة عينات مؤلفة من مُدخلات مقرونة بالخرج الملائم لكل مُدخل. ومع أن هناك العديد من البنى الحسابية المختلفة التي يمكن استخدامها، فإننا نركز هنا على الشبكات ذات الأوزان القابلة للتعديل. يتحقق التعلم بتكرار تعديل الأوزان في الشبكة إلى أن يصبح الأداء "حساب الفعل" مقبولا. وأتت تسمية الشبكات العصبونية **neural networks** لأنها تنمذج بعض خواص العصبونات الحية.

نتأمل الآن مسألة التعلم التالية:

لدينا مجموعة Ξ من متجهات **X** أبعادها **n** ومكوناتها x_i حيث **n ,... ,i=1**. يمكن لهذه المتجهات أن تكون متجهات السمات (يمكن أن تكون ترددات صوت في إشارة صوتية، أو قمم تحويل فورييه لها) التي يحسبها مكون المعالجة الإدراكية في الوكيل التفاعلي. وكما ذكرنا آنفا ، يمكن أن تكون قيم المكونات أعدادا حقيقية أو قيما بوليانية. ونعلم أيضا الفعل الملائم (معرفة الصوت من تردداته) **a**، لكل **X** في Ξ. يمكن أن تكون هذه الأفعال هي التي يرى المتعلمُ أن المدرس يقوم بها استجابة لمجموعة من المدخلات. تسمى هذه الأفعال المقرونة أحيانا لصيقات **labels** المتجهات أو صفوف **classes** المتجهات. تؤلف المجموعة Ξ إضافة إلى اللصيقات المقرونة بها ما يعرف بـ مجموعة التدريب **training set**. تكمن مشكلة تدريب الآلة في إيجاد دالة **f(X)** تستجيبُ لعناصر مجموعة التدريب "على وجه مقبول". ونهدف عادة إلى أن يتطابق الفعل الذي تحسبه الدالة **f** مع لصيقة أكبر قدر ممكن من المتجهات في Ξ. ونظرا لكون اللصيقات تعطى مع متجهات المُدخل نقول إن إجرائية التدريب مراقبة **supervised**.

ولكن، حتى لو وجدنا دالة تستجيب على نحو ملائم لمجموعة التدريب، فعلى أي أساس يرتكز اعتقادنا بأنها ستستجيب على نحو ملائم للمدخلات التي لم ترها خلال التدريب؟

إضافة إلى الدلائل التجريبية التي تدعم الاعتقاد بذلك، هناك مجموعة من النظريات التي تبين (بشروط معينة) أنه إذا كانت مجموعة التدريب "نموذجا" لأنماط المدخلات الأخرى التي يحتمل مصادفتها ، أي أنها تمثل تمثيلا جيدا كل فضاء المدخلات، فإن هذه المدخلات الأخرى تولد "على الأرجح" مخرجات "صحيحة تقريبا". في الواقع، هناك طرائق عديدة لتقدير الدرجة المحتملة لِضبط **accuracy** دالة **f** جرى تعلمها، عند تطبيقها على مدخلات مشابهة (لكن غير مرئية حتى الآن).

إن دراسة الشبكات العصبونية وتطبيقاتها تخرج عن نطاق هذا المقرر ولكن يمكن إعطاء أمثلة بسيطة تبين عملها.

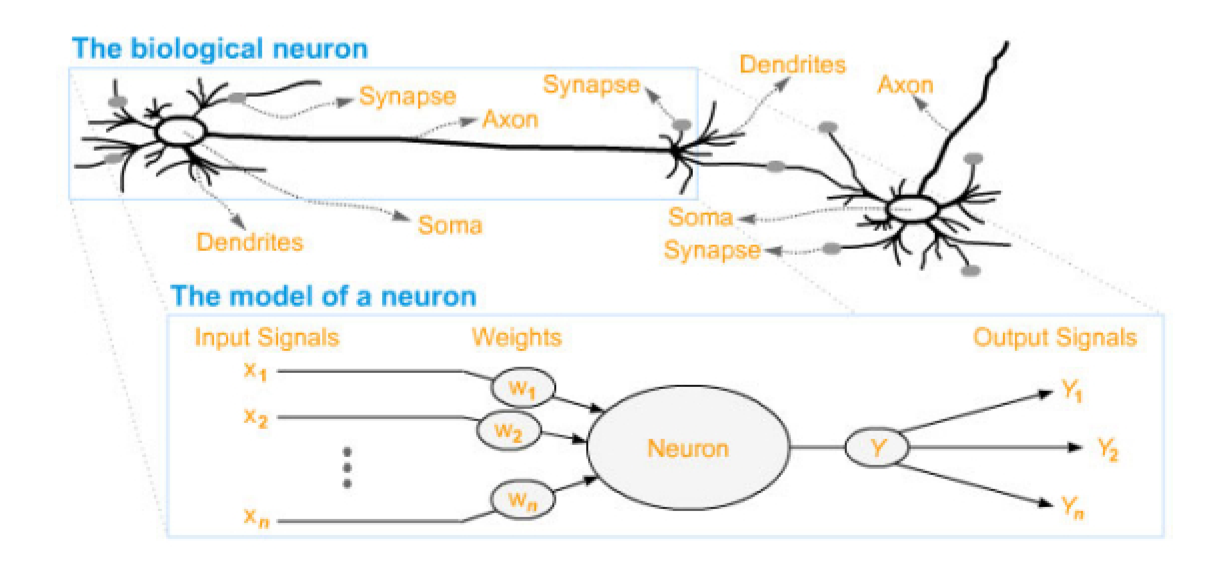
لنفترض أن لدينا عدة صور متساوية البعد كتب على كل منها رقم ما، بخط اليد. الشبكة العصبونية هي نظام يأخذ دخلا مجموعة سمات للصور (قد تكون بكسلات الرقم، وقد تكون تحويلات خاصة تحسب من الصورة)، وخرجه هو قرار يبين الرقم المدخل.

في مرحلة التدريب نمرر الصور الواحدة تلو الأخرى، ونعطي النظام الخرج الموافق الذي نرجوه لهذا النظام. كأن نقول عن كل صورة الرقم الموافق لها. تنفذ خوارزمية الشبكات العصبونية بحيث تغير أوزان الربط بين عقد الدخل (السمات) وعقد الخرج بحيث إذا أعيد إدخال الأمثلة تعطي المخرجات الموافقة. نقول إنه قد جرى تدريب الشبكة العصبونية بطريقة **supervised** (مع معلم لأنه يجب وجود شخص يحدد الخرج المناسب لكل دخل في مرحلة التدريب).

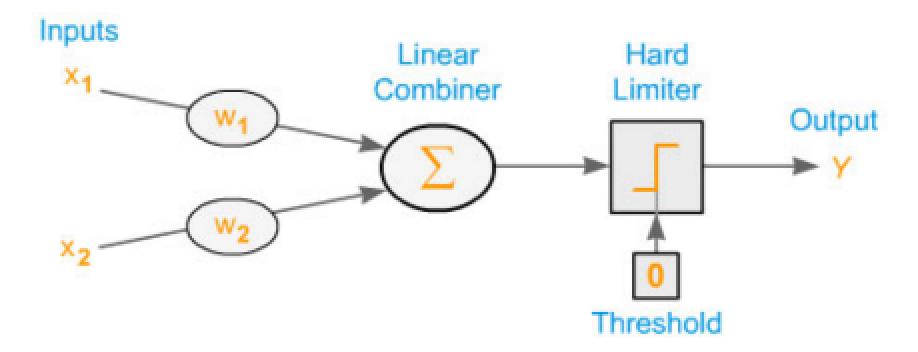
بعد ذلك، إذا أدخلنا صورة غير الصور التي دربنا النظام عليها، فسيكشف النظام (الشبكة العصبونية) المخرج الأقرب إليها. نقول إن الشبكة قد تعلمت الأرقام، وتستطيع تعرف أرقام من صور لم ترها من قبل.

في الحقيقة، تكمن إحدى الطرق بأخذ قياسات من المداخل وأخذ توزين لها ثم جمع النواتج وبحسب عتبة إما أن يهيج العصبون فيكون خرجه موجودا أو لا فلا يطابق الخرج الموافق.

يبين الشكل التالي كيف نحاول نمذجة عمل العصبونات الحيوية بعصبونات صنعية حسابية:


The biological neuron
Synapse
Axon
Synapse
Dendrites
Axon
Dendrites
Soma
Soma
Synapse
The model of a neuron
Input Signals
Weights
Output Signals
x1
x2
xn
w1
w2
wn
Neuron
Y
Y1
Y2
Yn

أما حل مسألة تصنيف بسيطة بخرج وحيد فيمكن أن تتم بعصبون له خرج وحيد يبينه الشكل التالي:



بهذه الفكرة، ترى هل بإمكاننا تعلم قواعد جديدة مفيدة في النظم الخبيرة؟ فيكون النظام قد اكتسب خبرة جديدة ومعرفة جديدة؟

نعتبر التعلم باستخدام الشبكات العصبونية من النوع الاستقرائي وذلك لأن الوظائف التي يجري تعلمها هي تخمينات (فرضيات) حول بعض الوظائف الهامة غير المعروفة. وفي تعلُّم ناجح، تعطي التخمينات، نموذجيا، مُخرجات صحيحة لمعظم المُدخلات الممكنة، ولكنها يُمكن أيضا أن تخطئ. تقوم طرائق التعلم الاستقرائي بإيجاد قواعد جديدة في مجال ما، لا يمكن اشتقاقها من أي قواعد سابقة. ويمكن أن ندرس بعض الطرق لتعلم القواعد استقرائيا في حساب الفرضيات والحساب الإسنادي.

OK	يجب الموافقة على القرض
COLLAT	ضمانة القرض الإضافية مقنعة
PYMT	طالب المال (العميل) قادر على سَداد دفعات القرض
REP	للعميل سمعة مالية جيدة
APP	تخمين الضمانة أكبر من مبلغ القرض بقدْر كافٍ
RATING	للعميل دفعات دورية منتظمة
INC	دخل العميل يتجاوز مصاريفه
BAL	للعميل نشرة موازنة ممتازة

3. التعلم الاستنتاجي عبر الأمثلة

يحسّن تعلمُ القواعد الاستنتاجي فعالية أداء النظام، وذلك باستنتاج قواعد جديدة من حقائق المجال وقواعده المعروفة سابقا. يُمكن أيضا اشتقاق النتائج التي يتوصل إليها النظام مستخدما هذه القواعد الإضافية، وذلك من دون هذه القواعد. ولكن بوجود هذه القواعد، يكون أداء النظام أشد فاعلية . سنشرح فيما يلي تقنية لاستنتاج القواعد الإضافية تسمى تقنية التعميم المبني على الشرح **Explanation-Based Generalization (EBG)**.

ولتأطير مناقشتنا، سنستخدم المثال البسيط لقبول القروض المصرفية. وهو كما يلي:

يُمكن أن نتصور، مثلا، موظف المصرف المسؤول عن منح القروض، وهو يستخدم نظاما كهذا

يُساعده ليُقرِّر: أمن المناسب منح قرض شخصي لأحد المتعاملين؟ قد يأخذ نظام القروض العملي، في الحسبان، الكثير من العوامل، أكثر بكثير مما يمكن أن نأخذه في المثال التوضيحي التالي. ولكن، يمكننا إعطاء فكرة تقريبية عن كيفية عمل مثل هذا النظام بتوصيف إصدار مبسط جدا. لنفترض أننا نقصد أن تشير الذرات في الدول إلى الفرضيات المرافقة لها:

وعلى هذا يُمكن استخدام القواعد التالية بغية اتخاذ القرار:

1. **COLLAT ∧ PYMT ∧ REP ⊃ OK**
2. **APP ⊃ COLLAT**
3. **RATING ⊃ REP**
4. **INC ⊃ PYMT**
5. **BAL ∧ REP ⊃ OK**

لنفترض الآن أن موظف القروض في البنك يريد أن يعرف النتيجة لعميل معين : أهي **OK** أم لا؟. لإثبات **OK**، يجب على محرك الاستدلال أن يبحث عن برهان في شجرة **AND/OR** مستخدما السلسلة الأمامية أو الخلفية (أو كليهما). سيكون لهذه الشجرة (إن وُجدت) عقدة **OK** باعتبارها عقدة الجذر، وحقائق (يحدد المستخدم أنها صحيحة أو هي موجودة في قاعدة بيانات حقائق النظام) باعتبارها عقد الأوراق. يرتبط الجذر بالأوراق (عادة، بواسطة عقد وسيطة) باستخدام القواعد.

باستخدام القواعد السابقة بنمط السلسلة الخلفية، يمكن البرهنة على **OK** هدف المستخدم إما بالبرهنة على **BAL** و **REP** معا **both**، أو بالبرهنة على **each** كلٍّ من **COLLAT** و **PYMT** و **REP**. تمثّل هاتان الطريقتان البديلتان للبرهنة على **OK** بالعقدتين **OK** الواقعتين تحت الجذر تماما. نذكّر أننا نسمي هذه العقد بعقد **OR**، في شجرة **AND/OR**. يُمكن البرهنة على عقدة خلفها عقد **OR** في شجرة **AND/OR**، بالبرهنة على أحد هؤلاء الخلف. وبتطبيق القواعد الأخرى، كما هو مُبين، تنتج مجموعات أخرى من العقد علينا أن نبرهن عليها.

لنفترض أنه عوضا عن أن تكون لدينا قواعد لهذه المسألة، ستكون لدينا مجموعة تدريب تتكون من قيم الواصفات لعدد كبير من الأشخاص (العملاء). ولتوضيح ذلك، لنتأمل المعطيات المعطاة في الجدول.

يمكن الحصول على هذا الجدول، مثلا، بالرجوع إلى سجلات طلبات القروض المصرفية والقرارات التي اتخذها موظفو المصرف بشأن قبول هذه القروض.

نسمي عناصر مجموعة التدريب التي تأخذ فيها **OK** القيمة **True** بالعينات الموجبة **positive instances**، والعناصر التي تأخذ فيها القيمة **False** بالعينات السالبة **negative instances**. ونريد، باستخدام مجموعة التعلم السابقة، استنتاج قواعد من الشكل:

$\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n \supset OK$

حيث α_i ذرات فرضية من المجموعة **{APP, RATING, INC, BAL}**.

إذا كان لمقدمة (الشروط القبلية) القاعدة القيمة **True** من أجل عينة معينة في مجموعة التعلم، قلنا إن القاعدة تشمل **cover** هذه العينة. يُمكننا تغيير أي قاعدة لنجعلها تشمل عددا أقل من العينات وذلك بإضافة ذرة إلى مقدمتها. مثل هذا التغيير يجعل القاعدة أكثر خصوصية **specific**. يُمكن لقاعدتين أن تشملا عددا من العينات أكبر مما تشمله قاعدة واحدة. ثم إن إضافة قاعدة، تجعل النظام الذي يستخدم هذه القواعد أكثر عمومية **more general**. ننشد مجموعة القواعد التي تشمل كل العينات الموجبة، دون غيرها، في مجموعة التعلم.

يُمكن أن تكون عملية البحث عن مثل هذه القواعد مكلفة حسابيا. سنشرح هنا طريقة " جشعة **greedy**" نسميها "فرق تسد **Separate and Conquer**". نحاول في هذه الطريقة، أولا، إيجاد قاعدة واحدة تشمل العينات الموجبة فقط -حتى لو كانت لا تشمل جميع العينات الموجبة. نبحث عن مثل هذه القاعدة بالبدء بقاعدة تغطي جميع العينات (الموجبة والسالبة)، ثم نجعلها بالتدريج أكثر خصوصية بإضافة ذرات لمقدمتها. ولأن قاعدة واحدة قد لا تكفي لشمول جميع العينات الموجبة، نضيف القواعد بالتدريج (بجعلها خصوصية بالقدر الذي نحتاج إليه) حتى الوصول إلى مجموعة كاملة من القواعد التي تشمل كل العينات الموجبة دون غيرها.

العميل	APP	RATING	INC	BAL	OK
1	1	0	0	1	0
2	0	0	1	0	0
3	1	1	0	1	1
4	0	1	1	1	1
5	0	1	1	0	0
6	1	1	1	0	1
7	1	1	1	1	1
8	1	0	1	0	0
9	1	1	0	0	0

معطيات المصرف

(نستخدم 1 لـ True و 0 لـ False)

لنر تطبيق هذه الطريقة على مثالنا . نبدأ أولا بالقاعدة المؤقتة التالية، التي تشمل جميع العينات: $T \supset OK$

يجب أن نضيف الآن ذرة لنجعلها تشمل عددا أقل من العينات السالبة، بالعمل باتجاه شمول العينات الموجبة فقط.

والسؤال المطروح هنا هو:

أي من الذرات التالية **{APP, RATING, INC, BAL}** يجب إضافتها؟

يوجد العديد من المعايير الممكن استخدامها للاختيار. وللإبقاء على مناقشتنا بسيطة سنجعل قرارنا يعتمد على النسبة، السهلة الحساب، التالية: $r_\alpha = n^+_\alpha / n_\alpha$

حيث n_α هي العدد الكلي للعينات (الموجبة والسالبة) التي تشملها المقدمة (الجديدة) للقاعدة بعد إضافة الذرة α إلى هذه المقدمة. أما n^+_α فهو العدد الكلي للعينات الموجبة التي تشملها المقدمة (الجديدة) للقاعدة بعد إضافة الذرة α إلى هذه المقدمة.

سنختار الذرة α التي تعطي أكبر قيمة لـ r_α.

تكون قيم النسب r_α في مثالنا هي:

$r_{APP} = 3/6 = 0.5$
$r_{RATING} = 4/6 = 0.667$
$r_{INC} = 3/6 = 0.5$
$r_{BAL} = 3/4 = 0.75$

ولذلك، سنختار الذرة **BAL**. مما يُعطي القاعدة المؤقتة: $\textbf{BAL} \supset \textbf{OK}$

تشمل هذه القاعدة العينات الموجبة **3** و **4** و **7**. إلا أنها تشمل العينة السالبة **1**، ولذلك يجب أن نخصصها أكثر.

سنستخدم التقنية نفسها لاختيار ذرة أخرى. وبالطبع، فإن حساب r_α الآن يجب أن يأخذ بالاعتبار أننا قررنا أن المكون الأول في مقدمة القاعدة هو **BAL**. لذا يكون لدينا:

$r_{APP} = 2/3 = 0.667 \quad r_{RATING} = 3/3 = 1.0 \quad r_{INC} = 2/2 = 1.0$

لدينا هنا تعادل بين **RATING** و **INC**. سنختار **RATING** لأنها تعتمد على عينة أكبر. (استكشف نتائج اختيار الذرة **INC** عوضا عنها).

تشمل القاعدة الجديدة $\textbf{BAL} \wedge \textbf{RATNG} \supset \textbf{OK}$ العينات الموجبة فقط، ولذلك، فلسنا بحاجة إلى إضافة ذرات جديدة إلى مقدمة هذه القاعدة. إلا أن هذه القاعدة لا تشمل جميع العينات الموجبة، ولا تشمل تحديدا العينة الموجبة **6**. ولذلك علينا إضافة قاعدة أخرى.

لتعلُّم القاعدة التالية، نقوم أولا بحذف جميع العينات الموجبة المشمولة بالقاعدة الأولى من الجدول للحصول على المعطيات المختصرة المبينة في الجدول الجديد. ونعيد الآن تطبيق الإجرائية ثانية بكمالها مع الجدول المختصر، بدءا من القاعدة $\textbf{T} \supset \textbf{OK}$ التي تشمل بعض العينات السالبة **1** و **2** و **5** و **8** و **9**. ولاختيار الذرة الواجب إضافتها إلى مقدمة القاعدة نحسب:

$r_{APP} = 1/4 = 0.25$
$r_{RATING} = 1/3 = 0.33$
$r_{INC} = 1/4 = 0.25$
$r_{BAL} = 0/1 = 0.0$

العميل	APP	RATING	INC	BAL	OK
1	1	0	0	1	0
2	0	0	1	0	0
3	1	1	0	1	1
4	0	1	1	1	1
5	0	1	1	0	0
6	1	1	1	0	1
7	1	1	1	1	1
8	1	0	1	0	0
9	1	1	0	0	0

معطيات المصرف

(نستخدم 1 لـ True و 0 لـ False)

نجد أن rRATING هي الأكبر، وهذا يعطي القاعدة $RATING \supset OK$.

تشمل القاعدة $RATING \supset OK$ العينات السالبة 5 و 9، ولذا يجب إضافة ذرة جديدة إلى مقدمتها . نحسب إذن النسب:

r_{RATING} = 1/2 = 0.5
r_{INC} = 1/2 = 0.5
r_{BAL} = 0/0

ونختار RATING وهذا يُعطي القاعدة:

$APP \wedge RATING \supset OK$

تشمل هذه القاعدة العينة السالبة 9. وبجعل هذه القاعدة أكثر خصوصية (باتباع الطريقة نفسها) تنتج أخيرا القاعدة التالية:

$APP \wedge RATING \wedge INC \supset OK$

تشمل القاعدتان المستنتجتان وهما:

$BAL \wedge RATNG \supset OK$

$APP \wedge RATING \wedge INC \supset OK$

جميع العينات الموجبة دون غيرها. وبهذا نكون قد انتهينا.

ولأن إجرائية إيجاد القواعد تستخدم بحثا جشعا (شرها)، يجب ألا يفاجئنا إمكان اختصار (تبسيط) قواعد التعلُّم أحيانا.

العميل	APP	RATING	INC	BAL	OK
1	1	0	0	1	0
2	0	0	1	0	0
5	0	1	1	0	0
6	1	1	1	0	1
8	1	0	1	0	0
9	1	1	0	0	0

المعطيات المختصرة

(تستخدم 1 لِ True و 0 لِ False)

يمكننا أن نختبر في حالة كل قاعدة، لنرى:

- أ بالإمكان حذف القاعدة دون تغيير القرارات المتخذة من قبل القواعد المتبقية بشأن مجموعة عينات التدريب؟ فإذا لم نجد أي تأثير (أو إذا وُجد تأثير ضعيف على الدقة عندما يكون ثمة ضجيج في المعطيات) أمكن حذف القاعدة.

وبالمشابهة، يمكننا أن نختبر في حالة كل ذرة في قاعدة لنرى:

- أبالإمكان حذف هذه الذرة وبتأثير أضعف؟

في الواقع، إذا كانت المعطيات كثيرة الضجيج، فقد نحتاج إلى تغيير المعيار الذي يقضي بأن تشمل القواعد المكتسبة (التي جرى تعلمها) جميع العينات الموجبة دون غيرها.

وعوضا عن ذلك، قد نسمح لكل قاعدة أن تشمل "في الدرجة الأولى" العينات الموجبة سامحين (كما يجب أن نفعل في حالة المعطيات الكثيرة الضجيج) بأن تشمل كل قاعدة عددا قليلا من العينات السالبة. وبالمشابهة، يُمكن أن نسمح لمجموعة القواعد المكتسبة أن تخفق في شمول بعض العينات الموجبة . تسمح عمليات "التشذيب" هذه، مع التعديلات التي تتحمل الضجيج (المتسامحة مع الضجيج) بالتخفيف من خطر التلبيق الزائد **Overfitting**.

يمكننا باختصار عرض إجرائية تعلم القواعد هذه، بالرماز المفترض **pseudocode**. ونسمي هذه

الخوارزمية بخوارزمية فرق تسد العامة **Generic Separate-and-Conquer Algorithm (GSCA)**. وفي هذه الخوارزمية:

- Ξ مجموعة عينات التدريب الابتدائية، لسمات بقيم اثنانية، لكل منها لصيقة بقيمة الذرة γ.
- π مجموعة القواعد المطلوب تعلُّمها.
- ρ إحدى القواعد، نتيجتها γ ومقدمتها (شرطها) Γ (عطف ذرات).
- α ذرة مشتقة من إحدى السمات في Ξ.

الرماز المفترض **pseudocode** لخوارزمية فرق تسد العامة **GSCA**:

1. استبدئ $\Xi \leftarrow \Xi cur$
2. استبدئ مجموعة قواعد خالية $\phi \leftarrow \pi$
3. كرر **repeat**. تضيف الحلقة الخارجية قواعد، حتى تشمل π جميع (أو معظم) العينات موجبة.
4. استبدئ $T \leftarrow \Gamma$
5. استبدئ $\Gamma \supset y \leftarrow \rho$
6. كرر **repeat**. تضيف الحلقة الداخلية ذرات إلى Γ، حتى تشمل ρ فقط (أو في الدرجة الأولى) عينات موجبة.
7. اختر ذرة α **choose**(α) تضيفها إلى Γ. (هذا الخيار غير حتمي، وقد يُستخدم للتعقب الرجوعي).
8. $\Gamma \wedge \alpha \leftarrow \Gamma$
9. حتى **until** تغطي ρ العينات الموجبة فقط (أو في الدرجة الأولى في **Ξcur**).
10. $\pi \leftarrow \pi$، ρ (نضيف القاعدة ρ إلى مجموعة القواعد π).
11. (العينات الموجبة في Ξcur والمشمولة بـ π) - $\Xi cur \leftarrow \Xi cur$.
12. حتى **until** تشمل π جميع (أو معظم) العينات الموجبة في Ξ.

يمكن استخدام حساب الإسناديات عوضا عن الفرضيات لتعلم قواعد جديدة، فيكون مثلا إذا تحققت إسنادية لعدد كبير من الأمثلة نقوم بالتعميم ونقول أن الإسنادية صحيحة مهما يكن المثال (استخدام المكمي العمومي).

4. الألعاب

يمكن استخدام التعلم في الألعاب لتعلم قواعد اللعب. ويدخل هذا النوع من التعلم ضمن طرائق التعلم بوجود معلم ويعتمد على الحصول على المعرفة المفيدة عبر إرشادات المعلم. يتكون نظام التعلم في هذه الحالة من المراحل التالية:

1. انتقاء التلميذ (برنامج التعلم) للعناصر التي تسمح ببناء هيكل مفهومي أولي يمثل مختلف المفاهيم النموذجية والاستدلالات المفيدة للوصول إلى السياق الخاص بالتعلم.
2. تمثيل العناصر المكتسبة إلى تمثيل داخلي في النظام بهدف التحليل أو غير ذلك.
3. تصفية موجهة للهيكل المفهومي الأولي وذلك بتحليل حالات عدم الترابط أو الإبهام أو النقصان المكتشفة فيه.

في أدبيات الذكاء الصنعي برمجة لعبة البوكر، وهي لعبة ذات معلومات ناقصة تعتمد على الحظ والحالة النفسية للاعب وهي بذلك تختلف عن لعبة الشطرنج ذات المعلومات الواضحة (الرقعة تبين تماما حالة اللاعب وخصمه). وقد برمجها **Waterman** في عام **1970**.

تتألف لعبة البوكر من خمس مراحل:

1. التوزيع: لكل لاعب خمس أوراق ويضع قطعة في إناء (لا يعرف كل لاعب سوى ورقه).
2. الرهان: بالتناوب على كل لاعب أن يتخذ أحد ثلاث قرارات:
   - الرهان: يضع مبلغا يساوي على الأقل آخر رهان للخصم،
   - الاستنكاف: ينهي سلسلة الرهان،
   - الانسحاب: ينتهي اللعب ويأخذ الخصم ما في الإناء دون أن يكشف عن أوراقه.
3. التبديل: يبدل كل لاعب **0** إلى **3** أوراق ويعلم الخصم عن عددها.
4. الرهان: كالسابق.
5. كشف الورق: من لديه نقاط أكثر بحسب الأوراق التي معه يربح ما في الإناء.

تعتبر عملية الخداع أساسية لأنها محاولة للربح بأوراق سيئة وذلك بإدهاش الخصم إلى أن ينسحب.

برمجة لعبة البوكر هي وضع قواعد لاتخاذ القرار بناء على تحليل الوضع وإقرار الفعل. مبدأ التعلم كما يلي:

- تعرض حالة لعب، يبين المعلم القرار المناسب اتخاذه والأسباب التي دعت لاتخاذ القرار.
- في حال توصل النظام الخبير إلى قرار مغاير (حسب مجموعة القواعد بداخله)، يجري تغييره سواء بإضافة قواعد جديدة أو تغيير قواعد سابقة لجعله يتوافق مع المعطيات التي أدلى بها المعلم.
- بعد عدة دورات مشابهة من التعلم يتقارب النظام الخبير إلى شكل يأخذ قرار المعلم نفسه في غالب الأحيان. تكمن الصعوبة في:
  - تقييم القرار المتخذ وفي حال كان سيئا معرفة القاعدة المسؤولة عنه.
  - تغيير القواعد الموجودة: وهنا نبحث عن قاعدة قابلة للتغيير (لها نفس جزء الشرط لقاعدة

التدريب) وقاعة قبل القاعدة الخاطئة (التي قادت إلى القرار الخاطئ) تماما فنغيرها بحسب المطلوب وإلا نبحث عن قاعدة قابلة للتغيير بعد القاعدة الخاطئة تماما فنغيرها وإلا نضيف قاعدة التدريب قبل القاعدة الخاطئة.

بهذا نكون قد أعطينا فكرة عن التعلم في الذكاء الصنعي واستخدامه في الألعاب.

5. مراجع البحث

- محاضرات في التعلم من مدرسة **ESIMAG** في غرونوبل **1992**.
- نوطة "الذكاء الصنعي **2**: التعلم الآلي" إعداد د. بسام الكردي. المعهد العالي للعلوم التطبيقية والتكنولوجيا **1995**.
- كتاب **artificial Intelligence: a guide to intelligent systems** للكاتب **Michael Negnevisky** الطبعة الثانية للناشر **Addison Wisley** عام **2002**.

الفصل السابع: التواصل والإدراك والفعل

1. مقدمة

رأينا في فصول سابقة كيف نبحث في فضاء الحالات عن حل لمسألة معينة. في هذا الفصل، سنفترض أن وكيلا (ربوطا) يستطيع القيام بعدد من الأفعال الأولية مثل التقاط غرض وتحريكه من مكان إلى آخر في بيئة تتضمن أغراض أخرى.

2. حلقة التحسس/التخطيط/الفعل

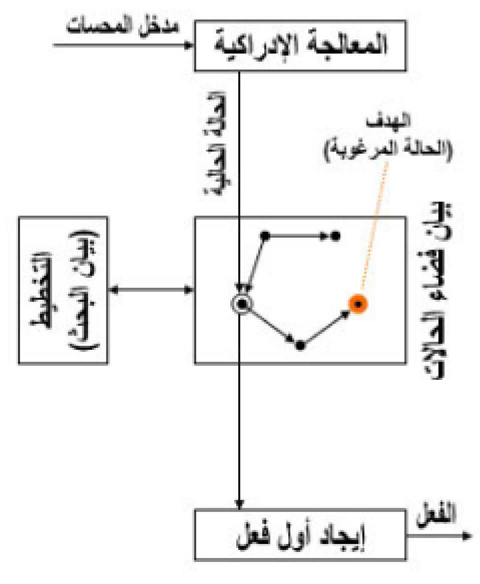
تعتمد فعالية طرائق التخطيط، المعتمدة على البحث، على عدة فرضيات متينة. غالبا ما تكون هذه الفرضيات غير محققة للأسباب التالية:

1. يمكن للإجرائيات المرئية ألا تعطي دوماً المعلومات الضرورية الخاصة بحالة البيئة (لأنها تحوي ضجيجا أو لا تتأثر بالخواص الهامة).
2. يمكن للأفعال ألا تحوي دوما تأثيراتها المنمذجة (لأن النماذج غير دقيقة بقدر كاف أو لأن نظام التأثير ينتج أحيانا أخطاء عند تنفيذ الأفعال).
3. يمكن وجود إجرائيات فيزيائية أخرى في العالم أو وكلاء أُخر، ويمكن لهذه الإجرائيات أن تغير العالم بحيث تتداخل مع أفعال الوكلاء.
4. مشكلة أخرى يسببها وجود التأثيرات الخارجية: يمكن للعالم أن يتغير خلال الزمن الذي يستغرقه بناء المخطط، بحيث لا يبقى المخطط مناسبا.
5. يمكن أن يطلب من الوكيل أن يقوم بفعل، قبل إتمام البحث عن حالة الهدف.
6. حتى لو كان لدى الوكيل الوقت الكافي، قد لا يسمح له مورده من ذاكرة الحساب متابعة البحث إلى حالة الهدف.

يوجد أسلوبان رئيسيان للتعامل مع مثل هذه الصعوبات للمحافظة على المميزات الأساسية للتخطيط المعتمد على البحث.

- الأول: باستعمال طرائق الاحتمالات لصياغة الارتيابات الإدراكية، والبيئية، وارتيابات المؤثرات.
- والآخر: نحاول العمل في محيط الصعوبات بإضافة عدة فرضيات وتقريبات مختلفة.

عوضا عن أن نلاحق هنا الطرائق المعتمدة على الاحتمالات، نقترح بنية، تسمى بنية تحسس/تخطيط/ فعل ، بحيث تتجاوز بعض التعقيدات المنتشرة بكثرة في العديد من



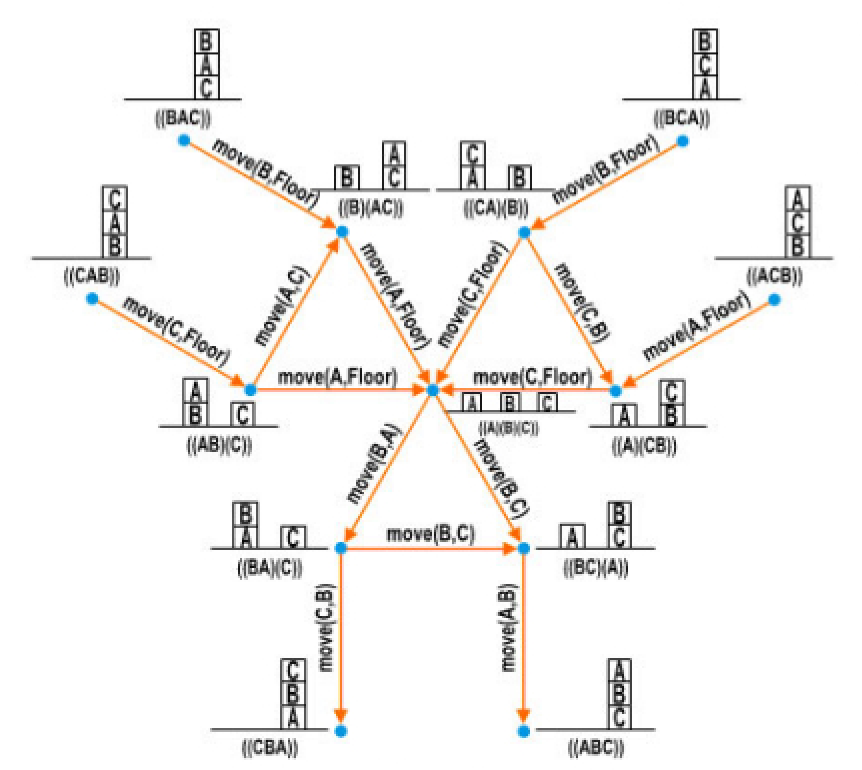
بنية وكيل تحسّس / تخطيط / فعل

التطبيقات. إن العقلاني في هذه البنية هو أنه حتى لو أنتجت الأفعال أحيانا تأثيرات غير متوقعة وحتى لو لم يستطع الوكيل أحيانا أن يقرر في أي حالة من العوالم هو، يمكن التعامل مع هذه الصعوبات تعاملا ملائما بجعل الوكيل يحصل على تغذية راجعة مستمرة من بيئته أثناء تنفيذ مخططه.

إحدى الطرائق لضمان التغذية الراجعة المستمرة هي التخطيط لسلسلة من الأفعال، ثم تنفيذ أول فعل فقط من هذه السلسلة، ثم تحسس حالة البيئة الناتجة، ثم إعادة حساب عقدة البداية، ثم تكرار الإجرائية. يقال عن الوكلاء التي تختار الأفعال بهذه الطريقة إنها وكلاء تحسس/تخطيط/فعل. على كل حال، ولكي تصبح هذه الطريقة فعالة، يجب ألا يتجاوز زمن حساب المخطط الزمن المخصص لكل فعل (انظر الشكل). في البيئات المعتدلة (تلك التي تتساهل ببعض الأخطاء)، "يحسب معدل" الأخطاء في التحسس والفعل على سلسلة حلقات تحسس/تخطيط/فعل.

تسمح التغذية الراجعة البيئية في حلقة تحسس/تخطيط/فعل في حل بعض الارتيابات الإدراكية والبيئية، وارتيابات المؤثرات. ومع ذلك، لكي تصبح هذه التغذية الراجعة فعالة، يجب أن نفترض أن التحسس والفعل هما -وسطيا- دقيقين. هذه الفرضية محققة في الكثير من التطبيقات. يمكن للوكيل غالبا أن يحسن دقة الإدراك وذلك بمقارنة المعطيات المتحسسة المباشرة بالنموذج المحفوظ للحالة الظاهرة.

تكمن إحدى الطرائق، لتقليل الشك ولتعويض النقص في معرفة الوكيل لتأثيرات أفعاله، في إجراء تغذية راجعة من البيئة باستمرار، ويمكن أيضا استخراج معلومات مفيدة من تجربة البحث ومن تجربة الأفعال في العالم. سوف نناقش الطرائق المختلفة التي يمكن فيها للوكيل أن يتعلم كيف يخطط ويعمل بفعالية أشد.



الأشجار الفرعية لمسألة تكديس الكتل

3. تعلم دوال تجريبية

إذا كان الوكيل لا يملك دالة تجريبية جيدة لكي يستطيع تقدير التكلفة إلى الهدف، يمكنه في بعض الأحيان تعلم مثل هذه الدالة.

سوف نشرح أولا إجراء تعلُم سهلا جدا يلائم الحالة التي يمكن فيها حفظ قائمة شاملة لجميع العقد الممكنة.

نفترض أولا أن لدى الوكيل نموذجا جيدا لتأثيرات أفعاله، وأنه يعلم أيضا تكاليف الانتقال من أي عقدة إلى عقد خلفها.

نبين في الشكل الأفعال الممكنة من الحالة الابتدائية **(BCA)** وكيفية الوصول إلى الحالة الهدف **(CBA)** في مسألة الكتل.

نستهل إجرائية التعلم بإعطاء القيمة **0** للدالة $\hat{h}$ قيمة ابتدائية لجميع العقد، ثم نبدأ خوارزمية بحث ***A**.
بعد توسيع العقدة n_i لتوليد الخلف $S(n_i)$، نعدل $\hat{h}(n_i)$ كما يلي:

$$\hat{h}(n_i) \leftarrow \min_{n_j \in S(n_i)} \left[\hat{h}(n_j) + c(n_i, n_j) \right]$$

حيث $c(n_i, n_j)$ هي تكلفة التحرك من n_i إلى n_j.

يمكن حفظ قيم $\hat{h}$ في جدول من العقد (لأنه لا يوجد منها سوى عدد قليل نسبيا). نفترض أيضا أنه إذا تم توليد عقدة هدف، ولتكن، n_g، فنعلم أن $\hat{h}(n_g) = 0$. ومع أن إجرائية التعلم هذه لا تساعدنا على الوصول إلى الهدف في المرة الأولى التي نبحث فيها، بأسرع من البحث ذي التكلفة الموحدة، فإن البحث المتتالي إلى الهدف ذاته (انطلاقا من حالات البداية المختلفة المحتملة) سوف يتسارع باستعمال الدالة $\hat{h}$ المتعلَّمة. وبعد عدة عمليات بحث، تنتشر تدريجيا، بالعودة إلى الوراء انطلاقا من عقد الهدف تقديرات أفضل فأفضل للدالة الحقيقية **h**.

أما إذا لم يكن لدى الوكيل نموذج جيد لتأثيرات أفعاله، فيمكنه تعلمها وتعلم دالة $\hat{h}$ في الوقت ذاته أيضا بإجرائية مشابهة، ولو أنه يجب تنفيذ عملية التعلم في العالم الواقعي عوضا عن نموذج فضاء الحالات. (بالطبع، يمكن لمثل هذا التعلم أن ينطوي على المصادفة!) نفترض أن الوكيل يملك بعض الوسائل لتمييز الحالات التي يزورها فعلا وأنه يمكنه تسميتها وبناء بيان شامل أو جدول لتمثيل الحالات مع قيمها المقدرة، والانتقالات الناتجة عن الأفعال. ونفترض أيضا أنه إذا لم يعلم الوكيل تكاليف أفعاله، فإنه يتعلمها بعد تنفيذها.

تبدأ الإجرائية بعقدة واحدة فقط، تمثل الحالة التي يبدأ منها الوكيل. ثم ينفذ فعلا ، يمكن أن يكون عشوائيا، فينتقل إلى حالة أخرى. وكلما زار حالة، سماها وأرفق معها قيمة لـ $\hat{h}$ كما يلي:

$$\hat{h}(n_i) \leftarrow \left[\hat{h}(n_j) + c(n_i, n_j) \right]$$

حيث n_i هي العقدة التي نفذ عندها الفعل **a** مثلا، وn_j هي العقدة الناتجة، و$c(n_i, n_j)$ هي تكلفة الفعل المكتشفة، و$\hat{h}(n_j)$ تقدير لقيمة n_j، والتي تساوي **0** إذا كانت n_j لم تزر سابقا قط وإلا تحفظ في الجدول.

كلما كان الوكيل على وشك تنفيذ فعل عند العقدة، **n**، التي حفظت عقد خلفها في البيان، اختار فعلا حسب السياسة التالية:

$$a = \underset{a}{\operatorname{argmin}}\left[\hat{h}(\sigma(n,a)) + c(n_i,\sigma(n,a))\right]$$

حيث $\sigma(n, a)$ هي وصف للحالة التي وصلنا إليها من العقدة **n** بعد تنفيذ الفعل **a**.

يبدأ إجراء التعلم الخاص هذا بالسير عشوائيا، مع احتمال العثور على هدف، وتنتشر قيم لـ $\hat{h}$ أفضل بالعودة إلى الوراء من جراء المحاولات المتتالية، مؤدية إلى مسارات أفضل. ليس من الضرورة تقويم جميع خلف العقدة **n** حين تحديث $\hat{h}(n)$، وذلك لأن الفعل المختار عند العقدة **n** هو الذي يقودنا إلى عقدة مقدرة تقع على المسار ذي التكلفة الصغرى من **n** إلى الهدف. لأننا نبني النموذج تدريجيا، من الممكن دمج "التعلم في العالم" في "التعلم والتخطيط في النموذج".

يمكن، مع ذلك، أن تنتج التقانة مسارات تعلم غير أمثلية لأنه من الجائز للعقد على المسارات الأمثلية ألا تكون قد زيرت قط. إن السماح بالأفعال العشوائية من حين لآخر (عوضا عن تلك المختارة من قبل السياسة المتعلّمة) يساعد الوكيل على تعلم مسارات جديدة (يمكن أن تكون أفضل) إلى الأهداف. يعتبر تنفيذ أفعال عشوائية إحدى الطرائق لتعامل الوكيل مع ما يسمى "الموازنة بين استكشاف (مسارات جديدة) واستعمال (المسارات المعروفة المتعلّمة سابقا)". من الجائز أيضا وجود طرائق أخرى للتحقق أن جميع العقد قد زيرت.

في بعض الأحيان، يكون بناء بيان شامل أو إنشاء جدول في جميع العقد مع الانتقالات فيما بينها غير عملي. عندما يكون لدينا نموذج لتأثيرات الأفعال (أي: عندما يكون لدينا مؤثرات تحول وصف حالة إلى وصف حالة خلف)، فإنه يمكننا تطبيق إجرائية بحث مقودة باستخدام دالة تقويم. نموذجيا، يمكن للدالة أن تعطي التقويم ذاته لعقد متعددة، لذا من الممكن تطبيق الدالة على وصف الحالة، على حين ليس الأمر كذلك بالضرورة فيما يخص حفظ جميع العقد وقيمها بشكل كامل في جدول.

نخمن أولا مجموعة من الدوال الجزئية التي نعتقد أنها يمكن أن تكون عناصر جيدة للدالة التجريبية. على سبيل المثال، في أحجية الثمانية يمكننا استعمال الدوال التالية:

W(n) = عدد المربعات التي هي في غير موضعها

P(n) = مجموع المسافات التي يبعد بها كل مربع عن "مكانه"

وأي دوال أخرى يمكن أن تتعلق بمقدار قرب الموضع من الهدف. ثم نكتب الدالة التجريبية باعتبارها تركيبا خطيا موزونا.

نكتب الدوال التجريبية للدوال المتعلقة بمقدار قرب الموضع من الهدف باعتبارها تركيبا خطيا موزونا كما يلي:

$$\hat{h}(n) = w_1 W(n) + w_2 P(n) + \ldots$$

يمكن أن نضبط $\hat{h}$ عند كل توسيع لعقدة. فبعد توسيع العقدة n_i لإنتاج عقد الخلف $S(n_i)$، نضبط الأوزان

بحيث:

$$\hat{h}(n_i) \leftarrow \hat{h}(n_i) + \beta\left(\min_{n_j \in S(n_i)}\left[\hat{h}(n_i,n_j) + c(n_i,n_j)\right] - \hat{h}(n_i)\right)$$

أو، بإعادة ترتيب العلاقة كما يلي:

$$\hat{h}(n_i) \leftarrow (1-\beta)\hat{h}(n_i) + \beta \min_{n_j \in S(n_i)}\left[\hat{h}(n_j) + c(n_i,n_j)\right]$$

حيث $0 < \beta \leq 1$ هو موسط معدل التعلم الذي يتحكم في مدى قرب $\hat{h}(n_i)$ من $\min_{n_j \in S(n_i)}\left[\hat{h}(n_j) + c(n_i,n_j)\right]$. عندما تكون $\beta = 0$، لا يجري أي تغيير البتة؛ وعندما تكون $\beta = 1$، نجعل $\hat{h}(n_i)$ تساوي $\min_{n_j \in S(n_i)}\left[\hat{h}(n_j) + c(n_i,n_j)\right]$. تؤدي قيم β الصغيرة إلى التعلم الشديد البطء، على حين يمكن لـ β بالقرب من 1 أن تجعل التعلم خاطئا وغير متقارب.

نلاحظ أنه يمكن أيضا تطبيق تقانة التعلم هذه حتى في الحالات التي لا يوجد فيها نماذج لتأثيرات الأفعال. هذا يعني، أنه يمكن تطبيق التعلم في العالم الواقعي كما شرحنا سابقا. نقوم بتنفيذ خطوة (يمكن أن تكون عشوائية أو مختارة حسب سياسة الفعل الناشئة)، وتقويم دالّتيها $\hat{h}$ قبل تنفيذها وبعده، ثم ملاحظة تكلفة الخطوة، وأخيرا إجراء تعديلات الأوزان.

4. الجوائز عوضا عن الأهداف

عند دراسة استراتيجيات البحث في فضاء الحالة، افترضنا أن للوكيل مهمة وحيدة قصيرة الأجل يمكن وصفها باستخدام شرط الهدف. وكان الهدف تغيير العالم إلى أن يحقق نموذجه (على شكل بنية معطيات) شرطا محددا.

في العديد من المسائل ذات الطابع العملي، لا يمكن صياغة المهمة بسهولة. عوضا عن ذلك، يمكن للمهمة أن تكون متغيرة باستمرار. يعبر المستخدم عن رضاه أو عدم رضاه عن أداء المهمة بإعطاء الوكيل جوائز **Rewards** إيجابية أو سلبية من حين لآخر. تصبح مهمة الوكيل الإكثار من كمية الجوائز التي يحصل عليها.

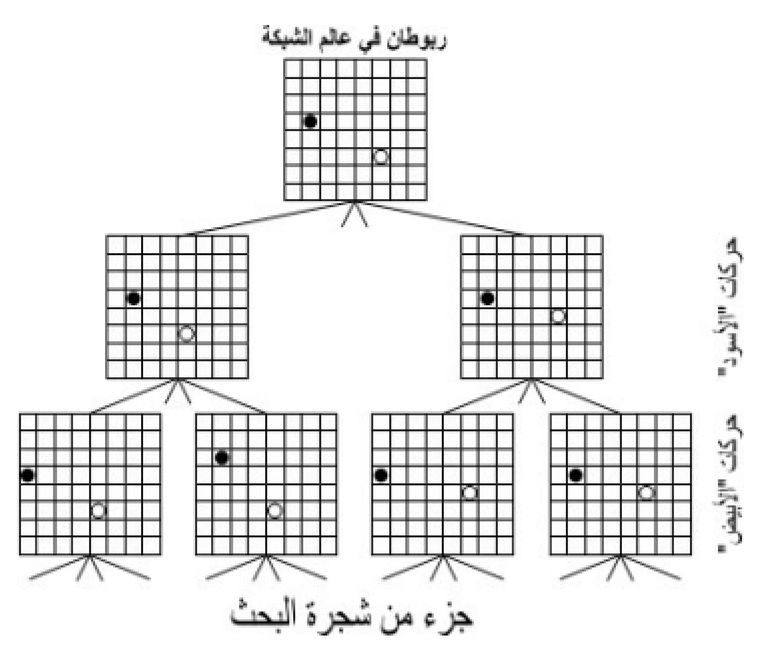
يمكن تحويل الحالة الخاصة التي هي مهمة بلوغ الهدف البسيطة في إطار العمل هذا بمكافأة الوكيل إيجابيا (مرة واحدة فقط) عندما يبلغ الهدف، وسلبيا (بمقدار تكلفة الفعل) في كل مرة ينفذ فعلا.
في هذا النوع من بيئة المهمات، نبحث عن توصيف لسياسة الفعل بحيث تكثر من الجوائز . وتكمن إحدى مشكلات الاستمرار في مهام غير منتهية، في الجوائز المستقبلية التي يمكن أن تكون لانهائية، وبذلك يصعب اتخاذ قرار بكيفية جعلها أكبر ما يمكن. إحدى الطرائق لمعالجة هذا، هي في تقليص الجوائز المستقبلية بعامل ما. بمعنى آخر، يفضل الوكيل الجوائز التي تتوفر في القريب العاجل على تلك التي تؤجل إلى المستقبل البعيد.

5. التخطيط في الألعاب ذات اللاعبين

تعد مسألة التخطيط والعمل ضمن بيئة مأهولة بأكثر من لاعب فعال من التحديات. يمكننا الذهاب قليلا إلى أبعد من استعمال بنية تحسس/تخطيط/فعل: التي لا تخطط في العمق ضمن مستقبل مجهول بسبب نقص في معرفة كيفية تصرُف المجموعة اللاعبة الأخرى. ومع ذلك، يمكن للاعب - عندما تتوفر المعرفة- بناء خطط تأخذ بالحسبان صراحة آثار أفعال مجموعة لاعبة أخرى. لنتأمل الحالة الخاصة للاعبين اثنين فقط. نجد في الوضع المثالي عندما يأخذ كل لاعب بالحسبان تصرف الآخر، أن دور كل لاعب يتخلله دور اللاعب الآخر بالتناوب (يتناوب اللاعبان اللعب). فيلعب اللاعب الأول، ثم الآخر، وهكذا دواليك.

ليكن، على سبيل المثال، في عالم شبكة الفضاء المبين في الشكل التالي ربوطان **Robots**:

يسمى الأول "أسود" والثاني "أبيض" يمكن أن يتحرك كل منهما إلى الخلية المجاورة في الصف أو العمود ذاته. هذان اللاعبان يلعبان بالتناوب (لنقلْ إن الأبيض سيبدأ)، ومن يجيء دوره، عليه أن يتحرك إلى أي مكان. لنفترض أن هدف اللاعب الأبيض أن يكون مع اللاعب الأسود في خلية واحدة، فيصبح هدف اللاعب الأسود أن يمنع ذلك من الحصول. يمكن للاعب الأبيض أن يخطط بناء شجرة بحث، نجد فيها أيضا -مع تناوب المستويات- كل حركات اللاعب الأسود المحتملة. يبين الشكل نفسه جزءا من شجرة البحث هذه.



جزء من شجرة البحث

وبهدف اختيار الحركة الأولى الفضلى، يحتاج اللاعب الأبيض إلى أن يحلل الشجرة كي يحدد أفضل النتائج، آخذا بالحسبان أن اللاعب الأسود سيلعب ليمنع اللاعب الأبيض من أن يصل إلى هدفه. وفي بعض حالات التضارب الخاصة بهذه الطريقة، من المحتمل أن يجد أحد اللاعبين حركة لا يهمه ما

يلعب الآخر بعدها، وبذلك يصل هذا اللاعب إلى هدفه. والأشيع، بسبب محدودية الحساب والزمن، لا يستطيع أي من اللاعبين أن يجد حركة تضمن له الربح. سوف نعرض طرائق البحث المحدودة العمق التي تفيد في إيجاد الحركات المعقولة تجريبيا في هذه الحالات. وفي جميع الأحوال، وبعد أن يختار الحركة الأولى، يقوم بها، ثم يراقب (يدرك) ما يفعله اللاعب الآخر، ثم يعيد إجرائية التخطيط على نمط تحسس/تخطيط/فعل.

إن المثال المختار: "عالم الشبكة" هو مثل على ما نسميه "ألعاب ذات لاعبين، كاملة المعلومات، ناتج الجمع صفر". لدينا في أحد الإصدارات، لاعبان، يلعبان بالتناوب كل بدوره إلى أن يربح أحدهما (ومن ثم يخسر الآخر)، أو تكون النتيجة تعادلا. لدى كل لاعب نموذج تام وكامل لبيئة اللعب ولجميع حركاته وحركات الخصم الممكنة وأثرها. (مع أنه لا يملك أي لاعب المعرفة التامة لما يمكن فعلا أن يلعبه الآخر بكل الحالات). تعطينا دراسة الألعاب من هذا النوع نظرة ثاقبة إلى بعض جوانب مسائل التخطيط التي هي أعم والتي تشمل عدة لاعبين، حتى لو لم يكن هنالك تضارب بين أهداف اللاعبين.

إن كثيرا من الألعاب الشائعة، ومنها لعبة الشطرنج، ولعبة الداما، ولعبة غو **Go**، تنتمي إلى هذا النوع من الألعاب. في الواقع، صممت برامج حاسوبية لتلعب مثل هذه الألعاب في مستويات عالية من المهارة في بعض الحالات. ومع ذلك فإن لعبة تيك تاك تو **Tic-Tac-Toe** ليست حاسوبيا باللعبة المثيرة، ولكنها تفيد ببساطتها في توضيح تقانات البحث. وهناك من الألعاب (مثل النرد، لعبة الطاولة) ما ينطوي على عنصر الحظ، مما يجعلها أكثر تعقيدا في تحليلها.

6. إجراء min-Max

نسمي اللاعبين من الآن فصاعدا **MAX** و**MIN**. وستكون مهمتنا إيجاد "أفضل" حركة للاعب **MAX**. لنفترض أن اللاعب **MAX** سيلعب أولا، ثم يلعب اللاعبان بالتناوب. وهكذا، تقابل مستويات العمق ذات الترتيب الزوجي المواقع التي يكون فيها على اللاعب **MAX** أن يلعب (دور **MAX**)؛ نسمي هذه العقد عقد **MAX**. وتقابل مستويات العمق ذات الترتيب الفردي المواقع التي يكون فيها على اللاعب **MIN** أن يلعب (دور **MIN**)؛ وهذه العقد هي عقد **MIN**. (علما أن العقدة العليا في شجرة الألعاب لها العمق صفر). وتحوي الطبقة ذات "عمق الطبقة **k**" في شجرة الألعاب العقد ذات العمق **2k** و**2k+1** . يحدَّد التوسع في البحث في أشجار الألعاب عادة باستخدام مصطلح عمق الطبقة: يقاس مقدار التقدم إلى الأمام بعدد أزواج الحركات المتناوبة للاعبين **MAX** و**MIN**.

كما أسلفنا سابقا، إن البحث الكامل (إلى الربح، أو الخسارة، أو التعادل) لبيانات معظم الألعاب هو أمر مستحيل حاسوبيا. فقد قدر بيان لعبة الشطرنج الكامل بنحو **1040** عقدة. وهذا يستغرق نحو **1022** قرنا لتوليد بيان البحث الكامل للعبة الشطرنج، حتى لو افترضنا أن توليد عقدة الابن تحتاج لـ **1/3** نانوثانية.

(مع العلم أن الكون قدر عمره بنحو **108** قرنا). أضف إلى ذلك أن تقانات البحث التجريبية لا تخفض عامل التفريع الفعلي بقدر كاف لكي يساعد كثيرا. لذلك، في حالة بعض الألعاب المعقدة علينا قبول أن البحث حتى النهاية هو أمر مستحيل (اللهم إلا عند الوصول إلى نهاية اللعبة). عوضا عن ذلك، علينا استعمال طرائق مشابهة لآلية البحث المحدودة الأفق المشروحة أعلاه.

يمكن أن نستعمل إما طريقة عرضا أولا، أو عمقا أولا، وإما الطرائق التجريبية **heuristic**، على أن نغير الآن شروط الانتهاء. يمكن أن نخصص عدة شروط انتهاء صنعية معتمدة على عوامل مثل حدود زمن البحث، أو حدود حجم التخزين، أو عمق أعمق عقدة في شجرة البحث. ومن المألوف أيضا في لعبة الشطرنج، على سبيل المثال، ألا ننهي اللعبة إذا كانت إحدى عقد الأوراق تمثل موقعا "مثيرا للنقاش"، فيستبدل به موقع ذو فائدة مباشرة.

بعد انتهاء البحث، علينا استنتاج تقدير لأفضل أول حركة من شجرة البحث. يمكن حساب هذا التقدير بتطبيق دالة تقويم سكونية على عقد الأوراق في شجرة الألعاب. تقيس دالة التقويم "قيمة" موقع عقدة ورقة. يعتمد هذا القياس على بعض خصائص مختلفة يتصوَّر أنها تؤثر في هذه القيمة؛ فعلى سبيل المثال، في لعبة الشطرنج، تقيس بعض الخصائص المفيدة ميزة الحجر النسبية، والتحكم في الوسط، والتحكم في الوسط من قبل الملك، وهلم جرا.

من المعتاد عند دراسة أشجار الألعاب أن نعتمد الاصطلاح الآتي:

- مواقع اللعب التي هي في مصلحة اللاعب **MAX** تؤدي إلى قيمة موجبة لدالة التقويم،
- على حين تؤدي المواقع التي هي في مصلحة اللاعب **MIN** إلى قيمة سالبة لدالة التقويم؛
- وأخيرا تقابل القيم القريبة من الصفر مواقع اللعب التي ليس من الواضح أنها في مصلحة أي من اللاعبين **MAX** أو **MIN**.

يستخلص الإجراء المسمى إجراء **minimax** أفضل أول حركة. (بغية التسهيل، نشرح هذا الإجراء وغيره من الإجراءات المتفرعة منه كما لو كان بيان اللعبة شجرة ألعاب حقيقية). نفترض أن اللاعب **MAX** يفضل العقدة ذات أكبر تقويم، عندما يكون عليه أن يختار بين عقد الأوراق في شجرة البحث. ونظرا لإمكان اختياره هذه العقدة بالفعل حين يأتي دوره باللعب، فإن القيمة المرجعة لعقدة **MAX** الأب لعقد الأوراق **MIN** تساوي القيمة العظمى للقيم السكونية لهذه العقد. من جهة أخرى، إذا كان على اللاعب **MIN** أن يختار بين عقد الأوراق، فإنه من المسلم به أن يختار تلك العقدة التي لها أصغر تقويم (يعني الأكثر سلبا). ونظرا لإمكان اختياره هذه العقدة حين يأتي دوره باللعب، فإننا نعطي العقدة **MIN** الأب لعقد الأوراق **MAX** القيمة المرجعة التي تساوي القيمة الصغرى للقيم السكونية لهذه العقد. وعند الانتهاء من إعطاء جميع الآباء لكل عقد الأوراق القيم المرجعة، نرجع قيم مستوى آخر، باعتبار أن **MAX** سيختار عقدة **MIN** التالية التي لها أعظم قيمة مرجعة، على حين سيختار **MIN** عقدة الخلف

MAX التي لها أصغر قيمة مرجعة.

نستمر بارجاع القيم، مستوى تلو الآخر بدءا من الأوراق، إلى أن، في النهاية، تعطى القيم المرجعة إلى خلف عقدة البداية. لقد افترضنا أن اللاعب **MAX** هو الذي يبدأ باللعب، لذا عليه أن يختار حركته الأولى تلك العقدة الخلف التي تقابل أعظم قيمة مرجعة.

يوضح مثالٌ بسيط طريقة "مينيماكس" وهو لعبة تيك تاك تو.

يتناوب في لعبة تيك تاك تو اللاعبان وضع علامة في جدول مصفوفة **3×3**. العلامة الأولى هي إشارة الضرب (×)، والثانية هي الدائرة (**O**). ويربح الذي يحصل أولا على صف كامل، أو عمود، أو قطر مملوء بعلامته.

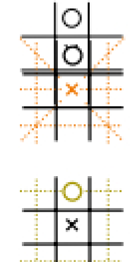
لنفترض أن إشارة الضرب (×) هي علامة اللاعب **MAX**، والدائرة (**O**) للاعب **MIN**، وأن على اللاعب **MAX** أن يبدأ أولا باللعب. نستعمل طريقة البحث عرضا أولا إلى أن نولد جميع عقد المستوى الثاني، ثم نطبق على مواقع هذه العقد دالة التقويم السكونية، وذلك إذا كان حد العمق هو **2**.

لتكن دالة التقويم **e(p)** للموقع **p** الذي نحصل عليه ببساطة كما يلي:

- إذا لم يكن الموقع **p** موقعا رابحا لأي من اللاعبين، فإن:
  e(p) = (عدد الصفوف والأعمدة والأقطار الكاملة التي بقيت مفتوحة للاعب **MAX**) – (عدد الصفوف والأعمدة والأقطار الكاملة التي بقيت مفتوحة للاعب **MIN**)
- إذا كانت **p** موقعا رابحا للاعب **MAX**، فإن: **e(p) = ∞** (نستعمل **∞** هنا للدلالة على عدد موجب كبير جدا)
- أما إذا كانت **p** موقعا رابحا للاعب **MIN**، فإن: **e(p) = -∞**

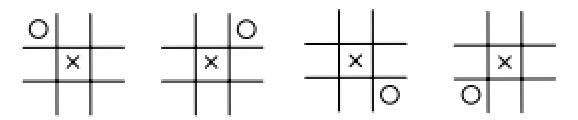
وهكذا إذا كانت **p** كما في الشكل الجانبي، يكون لدينا: **e(p) = 6 – 4 = 2**.

المفتوح لـ **MAX** القطران والعمودان الطرفيان وأسفل سطرين.



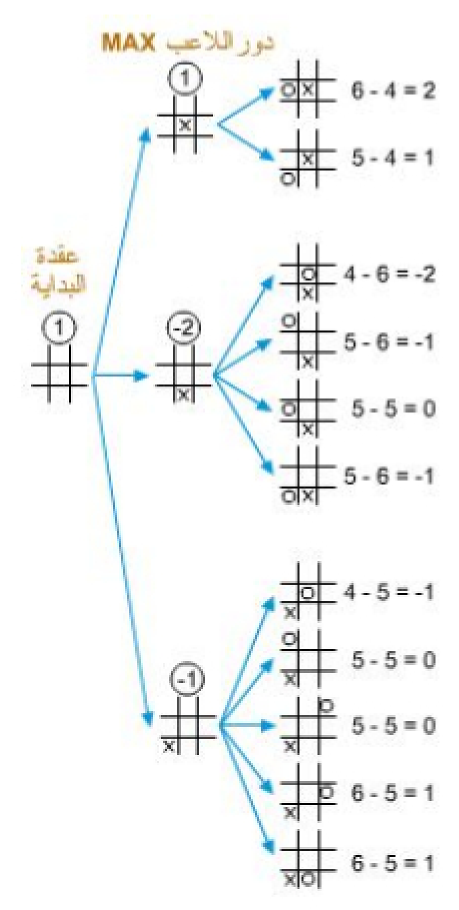
ما بقي مفتوحا لـ **MIN** هو العمودان الطرفيان والسطران الطرفيان.

ونستفيد من التناظر في توليد مواقع الخلف؛ لذا، جميع الحالات في اللعبة التالية تعد متطابقة:



وفي بداية اللعبة، يبقى عامل تفريع شجرة تيك تاك تو صغيرا بفضل التناظر؛ وفي نهايتها، يبقى صغيرا أيضا بفضل قلة عدد الخانات الفارغة المتبقية.

نبين في أشكال تالية الشجرة المولّدة بالبحث إلى العمق **2**. حيث نجد على يمين عقد الأوراق التقويم السكوني، والقيم المرجعة لباقي العقد محاطة بدائرة. ولأن أكبر لـ قيمة مرجعة، فسيجري اختيارها حركة أولى. وتتوافق هذه الحركة، على سبيل المصادفة، مع أفضل أول حركة للاعب **MAX** في حال إجراء بحث تام.

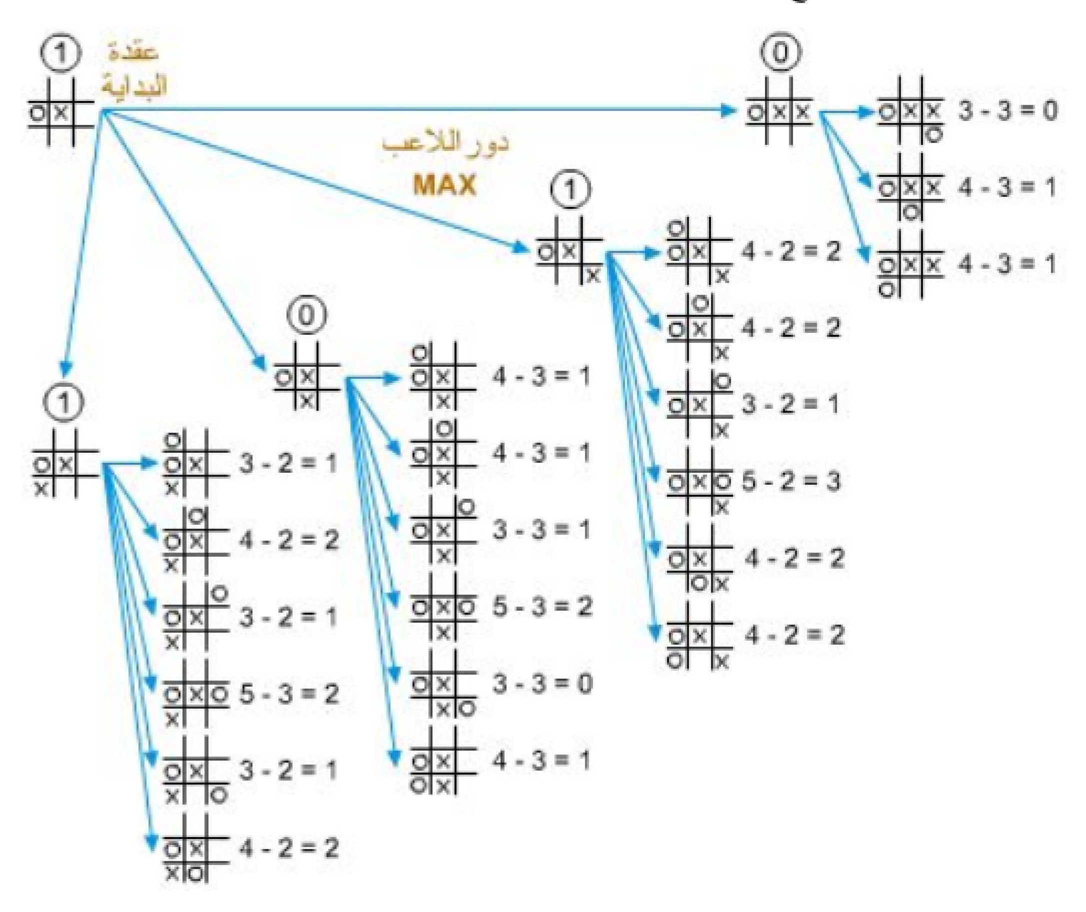


المرحلة الأولى من البحث
في لعبة تيك تاك تو

لنفترض الآن، متّبعين حلقة تحسس/تخطيط/فعل، أن اللاعب **MAX** قد لعب الحركة ورد اللاعب **MIN** بوضع دائرة في الخلية فوق إشارة الضرب تماما، وهي حركة سيئة للاعب **MIN**، الذي يظهر أنه لم يستعمل إستراتيجية بحث جيدة. يعطي البحث التالي للاعب **MAX** إلى العمق **2** أسفل التشكيلة الناتجة شجرة البحث المبينة في الشكل، إذ توجد الآن "أفضل" حركتين ممكنتين.

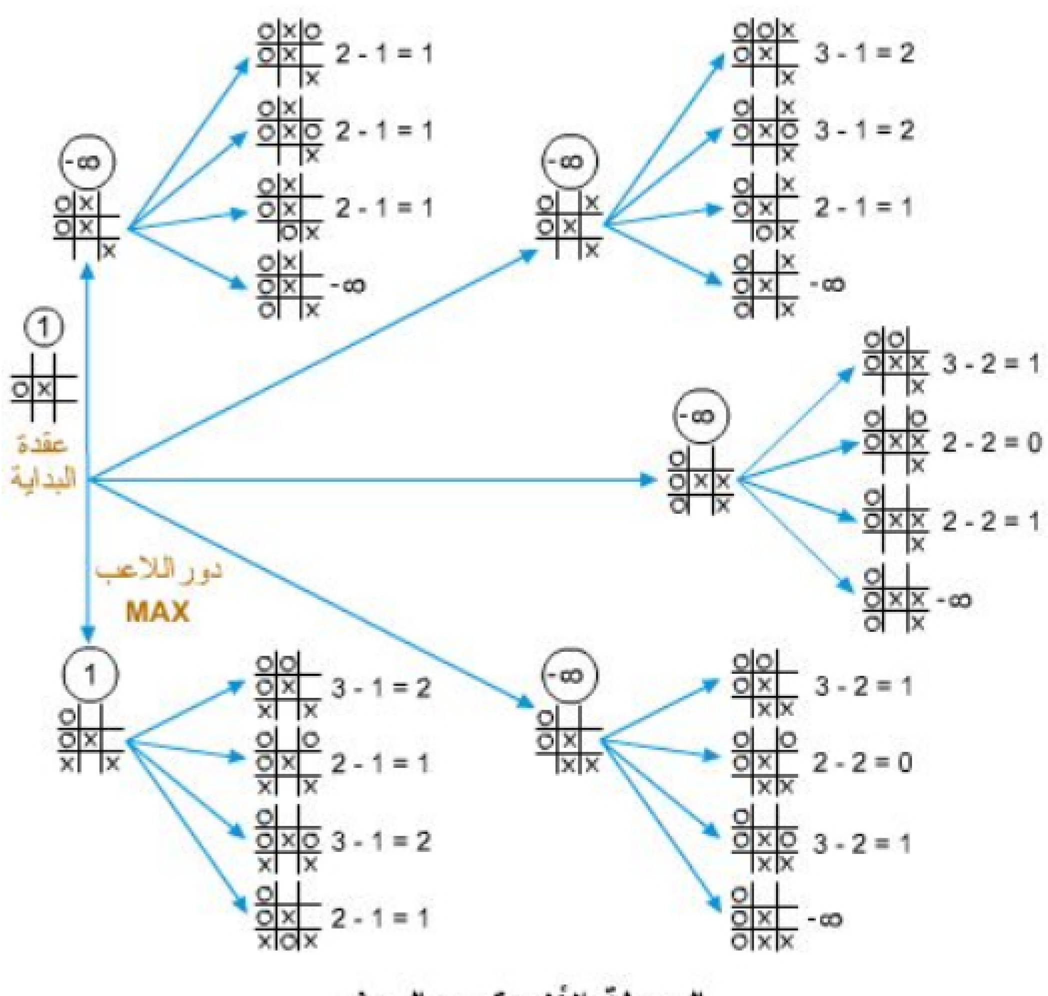
ولنفترض أن اللاعب **MAX** قد لعب تلك المشار إليها في الشكل. ويلعب الآن اللاعب **MIN** اللعبة التي

تجنبه الخسارة الحتمية، مما ينتج:



يتابع اللاعب **MAX** البحث، ليحصل على الشجرة المبينة، التي تشكل بعض عقد الأوراق فيه عقدا رابحة للاعب **MIN** ولهذا تقوم بـ ∞-. وعندما تحسب القيم المرجعة، نرى أن أفضل حركة للاعب **MAX** هي أيضا الحركة الوحيدة التي تجنبه الخسارة المباشرة. الآن يمكن للاعب **MIN** أن يدرك أن اللاعب **MAX** سيربح في حركته التالية، لذا يستسلم اللاعب **MIN** بلباقة.

يمكن أن يكون هذا البحث طويلا جدا لذلك يمكن تصغيره بوضع شروط على الحركات وهذا يقود إلى إجرائية تدعى ألفا-بيتا. يمكن دراستها في مرجع الذكاء الصنعي باللغة العربية.



المرحلة الأخيرة من البحث
في لعبة تيك تاك تو

7. مراجع البحث

- الذكاء الصنعي: رؤية جديدة

الفصل الثامن: استشراف المستقبل: الخوارزميات المتطورة

1. مقدمة

النظم الذكية هي نظم قادرة على حل المشاكل خلال زمن مقبول. على الإنترنت مثلاً يوجد عدد من النظم الذكية:

- وكلاء أذكياء:
  - تساعد على تصفح الوب، وإيجاد المعلومات ومطابقة المفردات التي نبحث عنها
  - ترشح البريد الإلكتروني
  - تنفذ إلى قواعد المعطيات، وتلخص المعلومات
  - تقوم بالتنقيب عن المعلومات باستخدام محركات بحث ذكية
  - تتصفح وثائق ذات حجم هائل
  - تراقب المعطيات وتحذر إذا حصلت بعض الأفعال
- يوجد أيضاً نظم خبيرة:
  - تطابق بين التساؤلات وإجابات على **FAQ Frequently Asked Questions**
  - تقوم بتصفح ذكي لقواعد معطيات بنوعيات مختلفة
  - تقوم بتصفح وثائق ذات حجم هائل

لقد تعرفنا في فصول سابقة على معظم تقنيات الذكاء الصنعي هذه: النظم الخبيرة، المنطق العائم، الشبكات العصبونية. لم نتعرف على تقنية الخوارزميات الجينية التي تعد حديثة نسبياً؛ فما هي هذه الخوارزميات وكيف تعمل؟

2. الخوارزميات الجينية

تسمى أيضاً بالحساب التطوري. فهل يمكن للتطور أن يكون ذكياً؟ وهل تتصرف الأعضاء الحية تجاه بيئتها بطريقة ذكية؟

يعتمد المنهج التطوري على نماذج حسابية مستقاة من الانتقاء الطبيعي والخوارزميات الجينية.

يتضمن الحساب التطوري: خوارزميات جينية و استراتيجيات تطور و برمجة جينية.

لن نتكلم عن أصل نظريات داروين في البقاء. سنتكلم عن الخوارزميات الجينية التي قدمها **John Holland** في سبعينيات القرن الماضي، وكان هدفه أن يقوم الحاسوب بما تقوم به الطبيعة. أما وصف هذه الخوارزمية فهو كما يلي:

- يتألف الصبغي الصنعي من سلسلة من الجينات يمثل كل منها بالبت **0** أو **1**.
- الطبيعة قادرة على التكيف والتعلم وإيجاد الصبغيات (الجينات) الجيدة على نحو أعمى.

➢ لدينا قياس لمدى ملاءة الصبغي للحفاظ على الإنتاج الذي يأتي نتيجة عمليتين:

- التزاوج:مبادلة جزأين من صبغيين (بداية الأول مع نهاية الثاني وبالعكس) (صورة **slide 20**)
- الطفرة: إجراء تغيير في مكان عشوائي من الصبغي. (صورة **slide 21**)

الخوارزميات الجينية الأساسية:

1. تمثيل نطاق متغيرات المسألة المطلوب حلها بصبغي طوله ثابت. نفترض عدد الصبغيات **N** واحتمال التزاوج p_c واحتمال الطفرة p_m.
2. تعريف تابع مواءمة يقيس ملاءمة الصبغي للحل.
3. توليد صبغيات عشوائياً عددها **N**. $x_1, x_2, \ldots x_N$
4. حساب ملاءمة كل من الصبغيات $f(x_1), \ldots f(x_N)$.
5. اختيار زوج من الصبغيات باحتمال يتعلق بملاءمتها.
6. توليد زوج جديد من الصبغيات بعد تطبيق التزاوج والطفرة.
7. وضع الصبغيات الجديدة في مجموعة جديدة
8. تكرار الخطوة (5) إلى أن تصبح المجموعة الجديدة بحجم **N**.
9. الاستعاضة عن مجموعة الآباء بمجموعة الأبناء.
10. العودة إلى الخطوة (4) والتكرار إلى أن يتحقق شرط توقف (منها إيجاد الحل).

مثال

لنفترض أننا نود إيجاد عدد بين **0** و **255** يكون عدد واحداته يساوي عدد أصفاره.

1. كل صبغي هو بايت بثماني بتات مثلاً العدد **h=126** يمثل بـ **01111110**. حجم مجموعة الصبغيات **N=4**.

h = | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | = 126

2. نعرف تابع الملاءمة $f(h) = 8 - |n_1 - n_0|$ القيمة الصغرى له هي **0** حين n_1 أو n_0 يساوي **8**، والقيمة العظمى **8** حين $n_1 = n_0 = 4$.
3. توليد **4** أعداد عشوائياً ولتكن: h_1, h_2, h_3, h_4 وتمثيلها هو :

0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	0
0	0	1	0	0	1	0	0

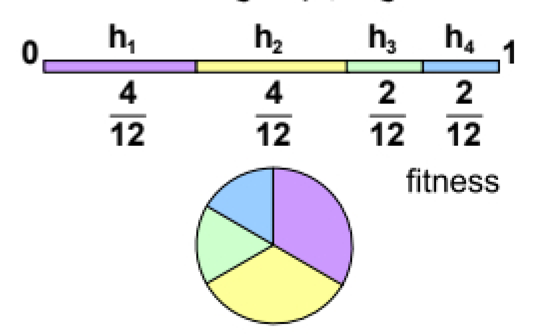
4. حساب توابع الملاءمة للصبغيات الأربعة $f(h_i)$ وهي:**4, 2, 4, 2**.

h_1	0	1	1	1	1	1	1	0	$f(h_1) = 8 - \|6 - 2\| = 4$
h_2	1	1	1	1	1	1	1	0	$f(h_2) = 8 - \|7 - 1\| = 2$
h_3	0	0	1	0	0	1	0	0	$f(h_3) = 8 - \|2 - 6\| = 4$
h_4	0	0	0	0	0	0	0	1	$f(h_4) = 8 - \|1 - 7\| = 2$

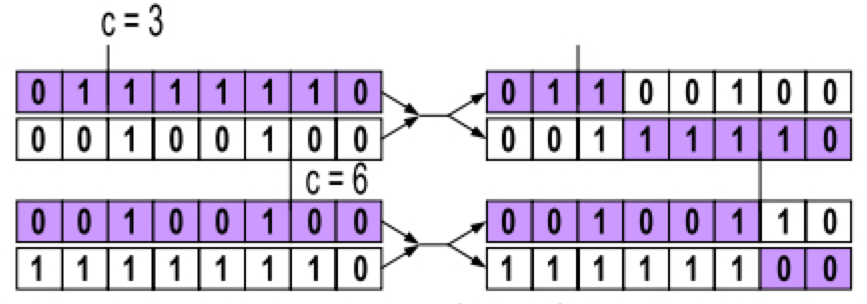
5. اختيار صبغيين بحسب احتمالاتها وملاءمتها $p(h_i) = \frac{f(h_i)}{\sum_{j=1}^{N} f(h_j)}$ فيكون:

$$p(h_1) = p(h_3) = 4/12; \quad p(h_2) = p(h_4) = 2/12$$

نقسم المجال **[0, 1]** إلى **N** جزء مقسمة بحسب **$p(h_i)$**، ثم نختار قيمة عشوائية $r \in [0,1]$ ونأخذ الصبغي الموافق (يمكن اختيار صبغي عشوائياً من بين الأربعة وكأنها على طرف دولاب)، ثم نختار صبغياً آخر.



6. إجراء المزاوجة والطفرة (نفترض أن الزوجين **h1, h3** و **h1, h2** وأن المزاوجة تتم في موضع البت **c = 3** للزوج الأول، وفي الموضع **c=6** للزوج الثاني، على سبيل المثال.



7. على الصبغيات الأطفال الأربعة نختار طفرة (تغيير بعض البتات)، ثم ندرس توابع المواءمة الجديدة فنحصل على القيم: **6, 8, 4, 6**. وهنا لا حاجة للتكرار لأن أحد الصبغيات ملائم ويحقق القيمة **8**.

0	1	1	0	0	1	0	0	$f = 8 - \|3 - 5\| = 6$
0	0	1	1	0	1	1	0	$f = 8 - \|4 - 4\| = 8$
0	0	1	0	0	1	1	0	$f = 8 - \|3 - 7\| = 4$
1	0	1	1	1	1	0	0	$f = 8 - \|5 - 3\| = 6$

كل تكرار للخوارزمية يسمى جيلاً، ويتراوح عدد الأجيال نموذجياً لمسألة بسيطة بين **50** و **500**.

قد يبقى أداء الأجيال مستقراً لفترة طويلة.

إذا لم نحصل على نتائج مرضية بعد عدد من الأجيال فإننا نعيد الخوارزمية من بدايتها.

3. البرمجة الجينية

البرمجة الجينية هي توسعة للخوارزميات الجينية تسعى لتطوير رماز برمجي يحل مسألة معينة (وليس مجموعة بتات).

يجري البحث في فضاء البرامج الحاسوبية الممكنة لحل مسألة ما.

كل برنامج هو سلسلة عمليات تطبق على معاملات.

المشكلة:

إيجاد لغة تقوم بإجراء تزاوج وطفرات على التوابع وعلى المعطيات. هذه اللغة يجب أن تسمح بمعاملة البرنامج على أنه معطيات وبتنفيذ المعطيات الجديدة على أنها برامج.

LISP هي اللغة المختارة لهذا الغرض، وهي لغة برمجة للقوائم نعتبر كل تابع هو قائمة رأسها التابع وجسمها محدداته.

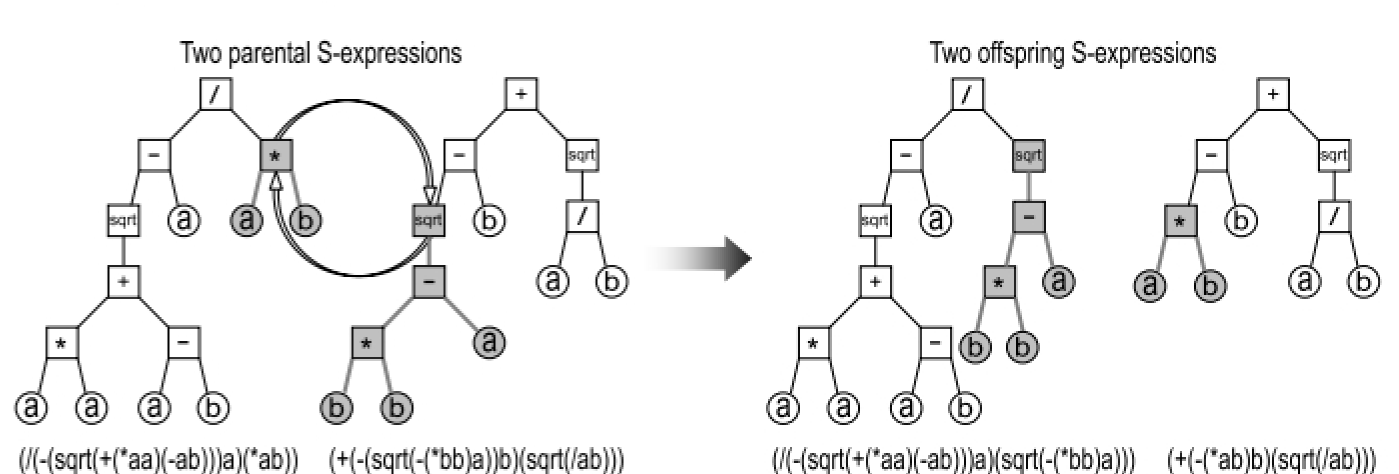
مثال:

(5 3)+ هي قائمة **list** رأسها + وجسمها **3** و **5**، وقد يكون أي عنصر فيها هو أيضاً قائمة.

يبين الشكل إلى اليسار التعبير **AB-C** الذي يكتب بالشكل: **(- (* A B) C)**

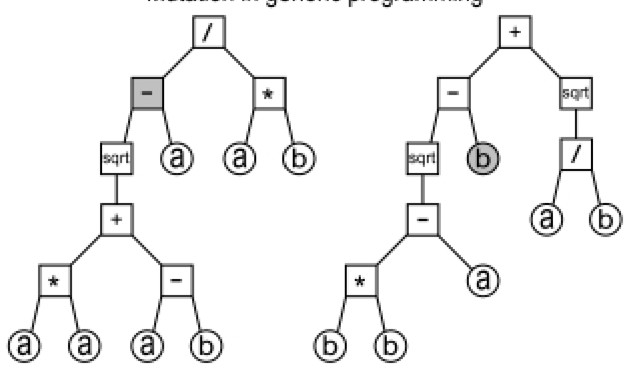
LISP S-expression (- (* A B) C)

وهنا عوضاً عن أن يكون الأبوان سلسلتي بتات سيكونان شجرتين مثل الشكل السابق، التزاوج بينهما هو تبديل بين فرعين من فروع كل شجرة كما في الشكل التالي:



في الشكل السابق، الصورة إلى اليسار تبين الأبوين، وإلى اليمين الطفلين الناتجين من التزاوج.

أما الطفرات فتظهر في العقد الرمادية من الشكل التالي:



4. نظم ذكية هجينة

رأينا أن المشاكل التي يتطلب حلها ذكاء صنعي هي على عدة أنواع:

- **التشخيص:** الاستدلال على سوء عمل غرض ما من سلوكه، وإيجاد حلول.
- **الاختيار:** اختيار أفضل خيار من قائمة خيارات.
- **التنبؤ:** التنبؤ بتصرف غرض في المستقبل من تصرفاته الماضية.
- **التصنيف:** إسناد غرض إلى صف معين معرف.
- **العنقدة:** تقسيم مجموعة غير متجانسة من الأغراض إلى مجموعات جزئية متجانسة.
- **الأمثلة:** تحسين نوعية الحلول حتى الوصول إلى حل أمثل.
- **التحكم:** التحكم بتصرفات غرض ليحقق متطلبات معينة بالزمن الحقيقي.

وفيما يلي سنجري مقارنة بين تقنيات الذكاء الصنعي التي جرت دراستها حتى الآن، استناداً إلى بعض النقاط الهامة.

مقارنة التقنيات الذكية

يمكننا الجدول التالي من المقارنة بين عدة تقنيات للنظم الذكية استناداً إلى بعض النقاط المهمة.

النظم المقارنة / نقاط المقارنة	النظم الخبيرة	النظم العائمة	الشبكات العصبونية	الخوارزميات الجينية
تمثيل المعرفة	فوق الوسط	جيد	سيء	تحت الوسط
تحمل معرف غير مؤكدة	فوق الوسط	جيد	جيد	جيد
تحمل معرف غير دقيقة	سيء	جيد	جيد	جيد
تكيف	سيء	تحت الوسط	جيد	جيد
إمكان التعلم	سيء	سيء		جيد

إمكان التوسع	جيد	جيد	سيء	تحت الوسط
كشف المعطيات والتنقيب عنها	سيء	تحت الوسط	جيد	فوق الوسط
الصيانة	سيء	فوق الوسط	جيد	فوق الوسط

والتوجه الحالي هو بناء نظم ذكية هجينة تضم عدداً من التقنيات بآن. قد يكون الهجين أسوأ أو أفضل من النظم غير الهجينة المكونة له، يعتمد ذلك على حسن اختيار كل جزء من النظام الهجين لتحقيق الأفضل.

من هذه النظم:

- نظم تجمع بين النظم العائمة والشبكات العصبونية: **neuro-fuzzy system**.
- نظم تجمع بين النظم الخبيرة والشبكات العصبونية: **neuro-expert system**.
- إن ضم المحاكمة الاحتمالية إلى المنطق العائم والشبكات العصبونية والحساب التطوري يشكل نواة فرع حديث بارز في مجال الذكاء الصنعي يسمى الحوسبة اللينة **soft computing** يهدف إلى بناء نظم قادرة على المحاكمة والتعلم في بيئة غير مؤكدة وغير دقيقة.

(H)

| ¬H).p(¬H)